

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

**BAKALÁŘSKÁ PRÁCE**

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

ELPOD: Automatický registrátor do katalogů a vyhledávačů na Internetu  
ELPOD: Automatic Register to Catalogs and Search Engines on the Internet

## Zadání bakalářské práce

Student:

**Tomáš Hejret**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**ELPOD: Automatický registrátor do katalogů a vyhledávačů na Internetu**  
**ELPOD: Automatic Register to Catalogs and Search Engines on the Internet**

Zásady pro vypracování:

1. Seznamte se s problematikou internetových katalogů, principů registrace, autorizace a další. Shromážděte seznam informací nezbytných pro registraci v těchto katalozích.
2. Zmapujte možnosti a technologie pro automatické nebo poloautomatické (vyžadující ověření, volbu kategorií pro registraci atd.) registrování do katalogů a vyhledávačů.
3. Proveďte analýzu návrh a implementaci aplikace, která bude shromažďovat informace o vyhledávačích a na základě definovaných pravidel bude provádět automatizovanou registraci.
4. Aplikace dále bude ověřovat existenci registrace v katalogu, sledovat kvalitu katalogu (pagerank, srnk, alexa) a upozorňovat na případné změny.
5. Aplikace dále bude umět sledovat vývoj (sílu webu) zadané adresy vzhledem k registraci v katalozích.
6. Funkčnost aplikace demonstujete na reálných příkladech. Systém musí být řešen univerzálně tak, aby bylo možno přidávat další katalogy případně provádět změny při procesu registrace.
7. Zhodnotte dosažené výsledky a navrhnete budoucí vylepšení.

Seznam doporučené odborné literatury:

Kubíček, M., Linhart, J., 333 tipů a triků pro SEO, CPress  
Janouch, V., Internetový marketing Prosaďte se na webu a sociálních sítích, CPress  
Clifton, B., Google Analytics, Podrobný průvodce webovými statistikami, CPress  
Kubíček, M., Velký průvodce SEO Jak dosáhnout nejlepších pozic ve vyhledávačích, CPress  
Sirovich, J., Darie, C., SEO v PHP Programujeme profesionálně, CPress

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## **Poděkování**

Tímto bych také chtěl poděkovat panu Ing. Radoslavu Fasugovi PhD. za odborné vedení při tvorbě této bakalářské práce a za poskytnutí seznamu katalogů, rozšiřující současnou databázi.

Dále mé poděkování patří panu Ing. Janu Romancenkovu ze společnosti Quest Software, který mi poskytl licenční klíč pro software Toad Data Modeler od této společnosti s platností po větší část doby vypracovávání této práce.

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dále prohlašuji, že nenesu odpovědnost za údaje, vložené uživateli a za dodržování smluvních podmínek jednotlivých serverů s katalogy či vyhledávači – odpovědnost za toto nesou sami uživatelé Registrátoru.

V Ostravě dne 4.5.2012



Tomáš Hejret

## **Abstrakt**

Tato bakalářská práce se zabývá zautomatizováním procesu registrace do vyhledávačů a katalogů na internetu za účelem zlepšení SEO. Práce se skládá ze dvou hlavních částí. První část se zaměřuje na teoretickou stránku. To představuje seznámení se s hodnocením webových projektů vyhledávači, na základě kvality, množství a rychlosti nárůstu počtu zpětných odkazů. Dále popisuje preregistrační a postregistrační aktivity, spojené se SEO, a princip provádění registrace, odlišnosti v postupech registrace do katalogů apod. Druhá část obsahuje návrh a popis implementace systému v aplikačním rámci Nette, popis funkcí, které systém nabízí a shrnutí dosažených praktických výsledků.

## **Abstract**

This bachelor thesis discuss about automating of process of registration into catalogs and search engines on the Internet to improve SEO. Thesis consists of two main parts. The first part is focused to theoretical side. It means introduction to the search engine's rating of web projects, based on quality, amount and growth rate of backlink count. It also describes preregister and postregister activities, interconnected with SEO, and principles of registration itself, differences in processes for the registration into catalogs, etc. The second part contains a proposition and a description of system implementation in Nette framework, description of the functions that the system offers and summary of the achievements and practical results.

## **Klíčová slova**

SEO, vyhledávače, katalogy, zpětné odkazy, pozice ve výsledcích vyhledávání, page rank

## **Key Words**

SEO, search engines, catalogs, back links, position in search results, page rank

## **Seznam použitých symbolů a zkratek**

AJAX	– Asynchronous JavaScript and XML
DB	– Databáze
DOM	– Document Object Model
ERD	– Entitně-relační diagram
GUI	– Graphical User Interface
HTML	– HyperText Markup Language
HTTP	– HyperText Transfer Protocol
IS	– Informační Systém
MS SQL	– Typ SŘBD
MVC	– Model View Controller
MVP	– Model View Presenter
MySQL	– Typ SŘBD
ORM	– Objektově-relační mapování
PHP	– PHP Hypertext Preprocessor
SEO	– Search Engine Optimization
SQL	– Structured Query Language
SŘBD	– Systém Řízení Báze Dat
URI	– Uniform Resource Identifier
URL	– Uniform Resource Locator

# Obsah

1 Úvod.....	9
2 Konkurenční produkty.....	10
2.1 Nástroj RegTool.....	10
2.2 Nástroj PARK.....	10
3 Princip metody zvýšení návštěvnosti.....	11
3.1 Co je to vyhledávač.....	11
3.2 Co rozumíme pod pojmem katalog.....	11
3.3 Zpětný odkaz na webový projekt a jak jej vytvořit.....	11
3.4 Proč registrovat do vyhledávačů.....	11
3.5 Proč registrovat do katalogů.....	12
4 Průběh SEO optimalizace získáváním zpětných odkazů.....	13
4.1 Jak registrace probíhá.....	13
4.2 Pozor na penalizaci.....	14
4.3 Odlišnosti v postupu registrace.....	14
4.3.1 Zadávání údajů v několika krocích.....	14
4.3.2 Výběr kategorie.....	14
4.3.3 Skrytá pole se statickou hodnotou.....	15
4.3.4 Skrytá pole s dynamickou hodnotou.....	15
4.3.5 Záchytný obrázek (captcha).....	15
4.3.6 Generování formuláře JavaScriptem.....	15
4.3.7 Morfující názvy formulářových polí.....	16
4.3.8 Kontrola referrera.....	16
4.3.9 Nutnost vytvoření uživatelského účtu.....	16
5 Analýza informačního systému.....	17
5.1 Specifikace požadavků na IS.....	18
5.1.1 Funkční požadavky.....	18
5.1.2 Nefunkční požadavky.....	18
5.2 Vstupy IS.....	18
5.3 Výstupy IS.....	18
5.4 Datová analýza.....	19
6 Implementace webové aplikace.....	20
6.1 Run-time prostředí, testování.....	20
6.2 Zvolený SRBD.....	20
6.3 Použité frameworky.....	20
6.3.1 Framework Nette.....	20
6.3.2 Databázový framework dibi.....	20
6.4 Knihovny PHP tříd a funkcí.....	21
6.4.1 Gridito.....	21
6.4.2 Simple HTML DOM Parser.....	21
6.5 Vývojové nástroje.....	21
7 Struktura.....	22
7.1 Struktura systému.....	22
7.2 Struktura GUI.....	23
8 Přístup řídicí vrstvy k perzistentní vrstvě a plnění databáze daty.....	24
8.1 ORM.....	24
8.2 Plnění databáze daty o katalogích a vyhledávačích.....	24
8.2.1 Parser vstupních dat na internetu.....	24
8.2.2 Parser vstupních dat v souborech CSV.....	25
8.3 Přidávání dalších katalogů a vyhledávačů - parser maker iMacros.....	25
8.3.1 Co vše dokáže parser z maker vytěžit.....	26

8.3.2 Princip činnosti parseru.....	26
9 Implementace funkcí IS.....	30
9.1 Zjištění použité znakové sady.....	30
9.2 Převod relativních adres na absolutní.....	30
9.3 Detekce použití enginu pro katalogy.....	30
9.4 Kontrolování aktivity katalogu či vyhledávače.....	31
9.5 Autentizace majitele (správce) webového projektu.....	31
9.6 Registrace webových projektů do katalogů.....	32
9.7 Ověření existence odkazu na webový projekt v katalogu.....	32
10 Proces registrace.....	33
11 Registrační strategie.....	35
11.1 Nepoužité strategie.....	35
11.1.1 Stránka s registračními daty, iframe uvnitř.....	35
11.1.2 Předvyplňování polí formuláře pomocí JavaScriptu.....	35
11.1.3 Vyžadováno vytvoření uživatelského účtu pro registraci.....	35
11.2 Všeobecné typy strategií.....	36
11.2.1 Základní typ – jednoduchý registrační formulář.....	36
11.2.2 Vícekroková registrace.....	37
11.2.3 Strategie s captchou.....	37
11.2.4 Strategie s výběrem kategorie.....	38
11.2.5 Jednoduchý registrační formulář, jež předchází výběr kategorie.....	38
11.2.6 Předvyplňování polí formuláře pomocí PHP.....	38
11.2.7 Generováním makra pro plug-in iMacros.....	39
11.3 Specializované typy strategií.....	40
12 Závěr.....	41
13 Literatura, zdroje dat a zdrojových kódů.....	42
14 Seznam příloh umístěných na CD.....	43



# 1 Úvod

Všichni, kdo provozují webové stránky, webovou aplikaci či prezentaci nebo cokoliv jiného, přístupného skrz síť internet, ať už je obsah zaměřen na jakékoliv téma, určitě chtějí, aby tento web navštěvovalo co nejvíce lidí. V tomto mohou být důvody různé - např. kvůli reklamě vlastních produktů a služeb, k prodeji reklamního místa, chtějí, aby se jejich sdělení co nejvíce rozšířilo nebo kvůli čemukoliv jinému.

Tady nastává problém - jak docílit oné relativně vysoké návštěvnosti, jak svůj web zviditelnit, představit světu a zvyšovat jeho návštěvnost. SEO zahrnuje několik různých metod, jak vylepšit návštěvnost. Jednu z těchto metod využívá nástroj pro elektronické podnikání Registrátor do katalogů a vyhledávačů na internetu (dále jen „Registrátor“), který část práce udělá za uživatele.

Důvodem, proč se zabývat zautomatizováním registrací je fakt, že v čím větším počtu katalogů se bude nacházet odkaz na webový projekt, tím větší bude efekt z hlediska SEO. Vyplňování stále stejných údajů do stovek či tisíců formulářů je pro člověka jednak únavné, ale také časově náročné.

Cílem této bakalářské práce tedy bylo prozkoumat možnosti a principy automatické a poloautomatické registrace do katalogů a vyhledávačů na internetu. Na základě těchto poznatků pak implementovat systém, který plně či částečně zautomatizuje proces registrace do různých katalogů a vyhledávačů na internetu. Výsledný systém, dostupný na <http://tomhejret.cz>, bude co nejsnadněji rozšiřitelný o nové katalogy a to na uživatelské, respektive administrátorské úrovni. Rozšiřitelnost bude možná také po programátorské stránce z pohledu zvětšování podílu automatizovaného procesu vůči nutnému zásahu obsluhy a také z pohledu schopnosti registrovat do nových typů katalogů.

## 2 Konkurenční produkty

Existuje několik konkurenčních nástrojů, řešících stejnou problematiku jako Registrátor. S následujícími konkurenčními produkty jsem se více či méně seznámil.

### 2.1 Nástroj RegTool

Jedná se o placenou desktopovou aplikaci (<http://apptools.cz/aplikace/regTool.aspx>), která pravděpodobně během registrování komunikuje se vzdáleným serverem. S tímto nástrojem jsem se seznámil po uživatelské stránce.

Tento nástroj umí poloautomatickou registraci - uživateli je načtena webová stránka (katalog), jako v běžném webovém prohlížeči a zde uživatel provádí registraci, nástroj předvyplní formulářová pole, které je schopen rozpoznat, zbylá pole musí doplnit uživatel a poté potvrdit odeslání formuláře. Výhodou je, že pokud není tento nástroj schopen některé z polí vyplnit, má uživatel možnost kromě ručního zadání údaje také využít některou z funkčních kláves, po jejímž stisknutí doplní do zvoleného pole příslušný údaj. Poté z levého panelu vybere jiný katalog, do kterého chce registrovat. V levém panelu může uživatel u každého katalogu označit průběh registrace do katalogu, čímž se mu naskytne možnost vygenerování reportu o registraci do souboru PDF. Nástroj RegTool také nabízí funkci exportu registračních dat ve formátu XML.

Jakožto desktopová aplikace ale vyžaduje instalaci a nenabízí v žádném z případů zcela automatizovaný proces registrace.

### 2.2 Nástroj PARK

Jedná se o placenou on-line aplikaci PARK „Polo-Automatická Registrace do Katalogů“ (<http://www.seo-starter.cz>). S tímto nástrojem jsem se seznámil přečtením uživatelského manuálu ([http://www.seo-starter.cz/doc/park\\_seo\\_starter.pdf](http://www.seo-starter.cz/doc/park_seo_starter.pdf)), prohlédnutím informací na webových stránkách a vyzkoušením demo verze pro registraci do deseti katalogů.

Tento nástroj umí poloautomatickou registraci - načte stránku katalogu, uživatel se musí „proklikat“ k začátku registrace do katalogu, poté nástroj předvyplní do formuláře zadané údaje a nakonec uživatel doplní zbylé údaje – obvykle kategorii a záchytný obrázek a kliknutím na tlačítko formulář odešle. Předvyplňování fungovalo u všech deseti povolených katalogů perfektně. Po skončení registrace do katalogu se uživateli zobrazí výsledek registrace. Pak si uživatel zvolí jiný katalog, nebo přejde k následujícímu v seznamu pomocí ovládacích prvků v hodní liště. V té se nacházejí také tlačítka pro označení úspěchu či neúspěchu registrace, čímž se uživateli naskytá možnost vygenerovat report o registraci do souboru PDF.

Ani tento nástroj však nenabízí v žádném z případů registrace zcela automatizovaný proces, obsahuje menší zásobu katalogů (několik stovek). Další omezení spočívá v počtu registrovaných webových projektů (resp. za každý projekt je nutné zvlášť zaplatit). S tímto souvisí také to, že po započetí registrace webového projektu nelze měnit jeho údaje – údaje se váží k faktuře za možnost využití tohoto nástroje uživatelem.

Další možností registrace přes server <http://www.seo-starter.cz> je využití placené registrace, u které provozovatel zajistí manuální registraci do katalogů. Ceny se pohybují v závislosti na požadovaném počtu katalogů, do kterých má být webový projekt zaregistrován.

### **3 Princip metody zvýšení návštěvnosti**

Do tzv. optimalizace pro vyhledávače (Search Engine Optimization = SEO) lze zahrnout rozmanitou skupinu aktivit pro zvýšení cíleného provozu, který přichází na daný webový projekt z vyhledávacích strojů [1]. V případě Registrátoru je tou aktivitou zvyšování počtu zpětných odkaz (tzv. linkbuilding), který patří mezi off-page faktory SEO, tedy faktory, které nelze ovlivnit na úrovni samotné webové stránky.

#### **3.1 Co je to vyhledávač**

Vyhledávač (internetový vyhledávač, vyhledávací stroj apod.) je internetová služba, jejíž součástí je mimo jiné robot, procházející webové stránky po celém internetu. Skrz hypertextové odkazy na jednotlivých stránkách se dostává na další podstránky či jiné weby a indexuje jejich obsah. Když pak uživatel zadá vyhledávací fráze pro vyhledávání v internetu, vrátí vyhledávač takové webové stránky, které nejvhodněji odpovídají vloženým frázím. Příklady internetových vyhledávačů jsou nejnavštěvovanější vyhledávač Google, Yahoo!, Bing nebo český Seznam.

#### **3.2 Co rozumíme pod pojmem katalog**

Katalog, přesněji katalog odkazů, je webový server, udržující databázi URL adres. Tyto adresy odkazují na různé webové projekty a tím zvyšují počet zpětných odkazů, vedoucích k určitému webovému projektu.

Některé katalogy nabízejí placenou a neplacenou variantu registrace. U takových katalogů při použití neplacené registrace se často zpětný odkaz v katalogu objeví se značným zpožděním. Naproti tomu placená registrace nabídne výskyt zpětného odkazu v katalogu v krátké době a mnohdy také registraci do dalších katalogů.

Dále se vyskytují katalogy s možností výměny odkazů – to znamená, že se v katalogu objeví odkaz na registrovaný webový projekt poté, co se odkaz na něj objeví v registrovaném webovém projektu. Alternativou k tomuto jsou katalogy, které výměnu odkazu nevyžadují, ale pokud je provedena, tak webový projekt s výměnou odkazů nějakým způsobem zvýhodňují např. zobrazováním odkazu na titulní stránce katalogu.

#### **3.3 Zpětný odkaz na webový projekt a jak jej vytvořit**

Registrátor je nástroj pro zvýšení počtu tzv. zpětných odkazů webového projektu – tedy hypertextových odkazů, umístěných na cizích doménách, směřujících na doménu webového projektu. Nejsnadnější cestou k vytvoření takového odkazu je registrování odkazu do katalogu odkazů.

#### **3.4 Proč registrovat do vyhledávačů**

Registraci do vyhledávače sice vždy nezíská webový projekt další zpětný odkaz, ale dříve se objeví ve výsledcích vyhledávání daného vyhledávače. Pokud registraci do vyhledávače neprovedete sami, musíte se spolehnout na to, až webový projekt najde robot (neboli crawler či spider) daného vyhledávače. Pokud nemá webový projekt zpětný odkaz, umístěný na silném webu, pak první indexování robotem a tím i první výskyt ve výsledcích vyhledávání může trvat i několik týdnů či měsíců. [1]

### **3.5 Proč registrovat do katalogů**

Jedním z kritérií, které ovlivní pozici webového projektu ve výsledcích vyhledávání internetových vyhledávačů je množství a kvalita zpětných odkazů. Kvalitní zpětné odkazy vznikají při umístění odkazu na web, který má stejně vysoké nebo vyšší vnitřní hodnocení relevantnosti webové stránky (v tomto případě katalogu), přidělené vyhledávačem. Toto vnitřní hodnocení stránek je označováno jako „page rank“ od Googlu, „srank“ od českého vyhledávače Seznamu apod. Případně velké množství slabých zpětných odkazů může doplnit nebo částečně vyvážit silný zpětný odkaz.

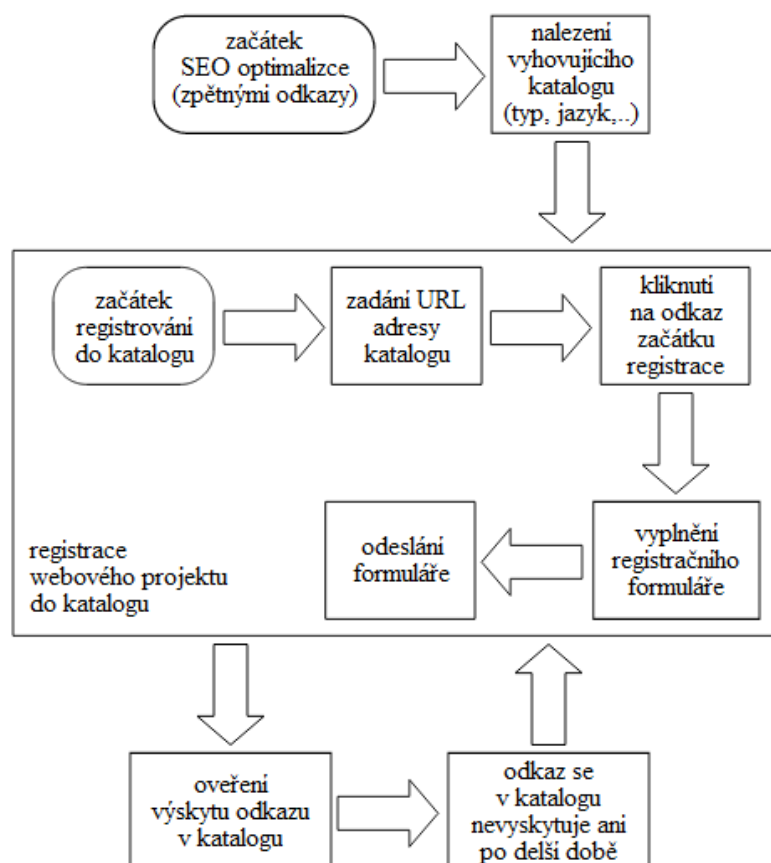
Katalogy, na rozdíl od vyhledávačů, nepoužívají automatické procházení internetu pomocí robotů. Přidávání odkazů je tedy závislé pouze na invenci ze strany provozovatele webového projektu. [1] Obvykle jsou registrované webové projekty ručně kontrolovány administrátorem katalogu.

## 4 Průběh SEO optimalizace získáváním zpětných odkazů

Pro úspěšné zaregistrování webového projektu do katalogu musí být provedena určitá posloupnost kroků. Principiálně je takový postup pokaždé stejný, ale prakticky se liší v mnoha ohledech (viz. kapitola 4.3 a podkapitoly).

### 4.1 Jak registrace probíhá

Opakováním procesu, zjednodušeně popsaného v diagramu na obrázku 1, lze získat zpětné odkazy registrací do katalogů.



Obrázek 1: Schéma postupu při SEO optimalizaci získáváním zpětných odkazů

Za blokem „vyplnění registračního formuláře“ (obrázek 1) se vlastně skrývá rozpoznání, jaká data do pole formuláře vyplnit, což člověk provádí intuitivně a to podle popisku vedle formulářového pole. V každém z automatizovaných postupů je nutné nejprve rozlišit, které údaje o registrovaném webovém projektu patří do kterého prvku (vstupního pole) formuláře. Tyto informace by se v databázi uchovávaly ve tvaru (zjednodušeně): název\_pole – typ\_údaje.

## 4.2 Pozor na penalizaci

Je nutné brát v potaz, že prudký nárůst počtu zpětných odkazů může být vyhledávači penalizován. Vyhledávač totiž primárně počítal s tím, že spokojení uživatelé webového projektu umístí odkaz na svůj blog, webové stránky apod. a tímto způsobem se budou pozvolna vytvářeny zpětné odkazy. Z velkého množství zpětných odkazů by tedy měla plynout velká obliba webového projektu a proto vyhledávač upřednostňuje takový projekt ve výsledcích, než podobné weby. Vytváření umělých (katalogových) odkazů je však v režii provozovatele webového projektu, který má snahu získat co nejlepší pozici ve vyhledávání v co nejkratším čase. Dalším způsobem získávání velkého množství zpětných odkazů v krátkém čase je založení několika fiktivních webů s příslušnými odkazy, což vyhledávače striktně označují jako nekalou praxi.

Pokud se chceme vyhnout penalizaci z rychlého nárůstu počtu zpětných odkazů, ale potřebujeme rychle zlepšit pozici ve výsledcích vyhledávání zpětnými odkazy, můžeme se ubrat cestou vybírání pouze „silných“ katalogů, které mají u vyhledávačů dobré hodnocení. Takové katalogy ale bývají často placené, nebo alespoň vyžadují výměnu odkazu.

## 4.3 Odlišnosti v postupu registrace

Pro úspěšné zaregistrování webového projektu do katalogu musí být provedena určitá posloupnost kroků. Principiálně je takový postup pokaždé stejný, ale prakticky se liší v mnoha ohledech.

Bohužel téměř každý katalog (všechny katalogy na bázi některého z enginů pro katalogy lze brát z tohoto hlediska jako jeden) je unikát a má nějakou odlišnost od základního postupu, případně i navíc některý z typů zabezpečení proti automatizovaným strojům. Navrhnout systém, který by dokázal všechny níže uvedené metody a jejich různé kombinace zpracovat, by se musel vůči katalogu chovat jako člověk, pro kterého byla primárně registrace určena.

### 4.3.1 Zadávání údajů v několika krocích

Některé katalogy provádějí registraci v krocích, kdy v se každém kroku vyplňuje pouze část ze všech údajů, které budou během registrace zadávány. Po zadání údajů v dílčím kroku se data odešlou serveru, který je zvaliduje a buď oznámí, že některé z dat je zadáno nesprávně (např. chybný tvar URL adresy) nebo přesměruje k dalšímu kroku. Obvykle se data ze všech kroků odesílají na stále stejnou URL. V takovém případě by opět stačilo zaslat data serveru v podobě `název_pole1=hodnota_pole1&název_pole2=hodnota_pole2&....název_poleN=hodnota_poleN`

### 4.3.2 Výběr kategorie

Katalogy se liší jednak v množství kategorií a také zda mají i podkategorie. Následně se liší ve způsobu výběru kategorie:

- formulářový prvek SELECT
- dva prvky SELECT, první pro výběr kategorie a druhý pro podkategorii, kterému je seznam hodnot načten JavaScriptem po vybrání kategorie
- přepínače
- textové pole (někdy doplněné tlačítkem, po jehož stisku se zobrazí seznam přípustných hodnot)

- registrace do právě procházené kategorie odkazů (nejprve zvolit kategorii, poté v dalším kroku vyplnit formulář)

### 4.3.3 Skrytá pole se statickou hodnotou

Formuláře běžně obsahují skrytá pole, jejichž hodnota nelze za běžných okolností uživatelem změnit ani pro něj není viditelná. Tyto hodnoty využívá skript, který vyhodnocuje formulář buď k rozlišení o který formulář se jedná (pokud jich zpracovává více), nebo jako základní prvek zabezpečení. Tato pole je potřeba nalézt a v případě, kdy lze zasílat vyhodnocovacímu skriptu data (zcela automatizovaný postup), si uchovávat jejich hodnotu, aby nebylo nutné opětovně načítat webovou stránku a tento údaj hledat. V případě předvyplňování není extrahování takové hodnoty nutné.

### 4.3.4 Skrytá pole s dynamickou hodnotou

Některé formuláře naproti tomu generují hodnotu skrytého pole jako nějaký identifikátor relace s omezenou časovou platností jakožto zabezpečovací prvek. Protože musejí být tyto hodnoty vždy odeslány, tak v případě, kdy lze zasílat vyhodnocovacímu skriptu data (zcela automatizovaný postup), musí být načtena daná webová stránka a údaj extrahován z jejího zdrojového kódu. V případě předvyplňování není extrahování takové hodnoty nutné.

### 4.3.5 Záchytný obrázek (captcha)

Záchytný obrázek (příklady na obrázku 2) obsahuje číslice a písmena (která někdy tvoří slova) s různými velikostmi, barvami, fonty písma, jejich deformací či různé rušivé symboly, umístěné přes text, různá pozadí textu na záchytném obrázku atd. Tento bezpečnostní kód musí být správně opsán do příslušného pole formuláře, jinak nebude formulář dále zpracován.



Obrázek 2: Příklady záchytných obrázků

### 4.3.6 Generování formuláře JavaScriptem

Pokud se o vytvoření formuláře stará JavaScript, není možné ve zdrojovém kódu HTML stránky nalézt formulářové prvky.

Pokud bychom i v takovémto případě požadovali zcela automatickou registraci, musel by být implementován stroj, interpretující jazyk JavaScript. Díky němu by bylo možné nalézt pole formuláře a extrahovat jejich názvy. Před postupem registrace by se muselo opět uvážít, jaké další odlišnosti od základního postupu tato registrace obsahuje.

### 4.3.7 Morfující názvy formulářových polí

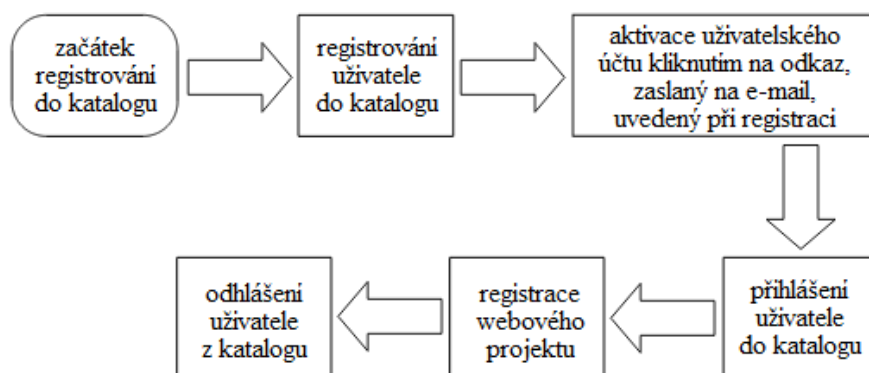
Formulářová pole dostanou po každém načtení či obnovení webové stránky s formulářem nový název a identifikátor není uveden vůbec. Poměrně málo používaná, ale velice nepříjemná technika zabezpečení, která znemožňuje využití seznamu názvů polí, uložených v DB.

### 4.3.8 Kontrola referrera

Referrer je v podstatě předchozí navštívená URI, ze které uživatel přišel na URI, která referrera kontroluje, což patří mezi prvky ochrany proti sociálnímu inženýrství a tím pádem proti automatizovaným strojům.

### 4.3.9 Nutnost vytvoření uživatelského účtu

Další prvek zabezpečení katalogů, se kterým se lze setkat, je omezení v přidávání odkazů pouze zaregistrovaným a přihlášeným uživatelům. Registrování uživatele se provádí vyplněním formuláře, odlišného od toho pro registraci webového projektu. Po vyplnění a odeslání formuláře obvykle přichází na zadaný e-mailový účet e-mail s potvrzovacím odkazem. Po kliknutí na odkaz je účet uživatele aktivován a ten se již může přihlásit a registrovat webové projekty do katalogu. Toto schéma postupu znázorňuje diagram na obrázku 3.



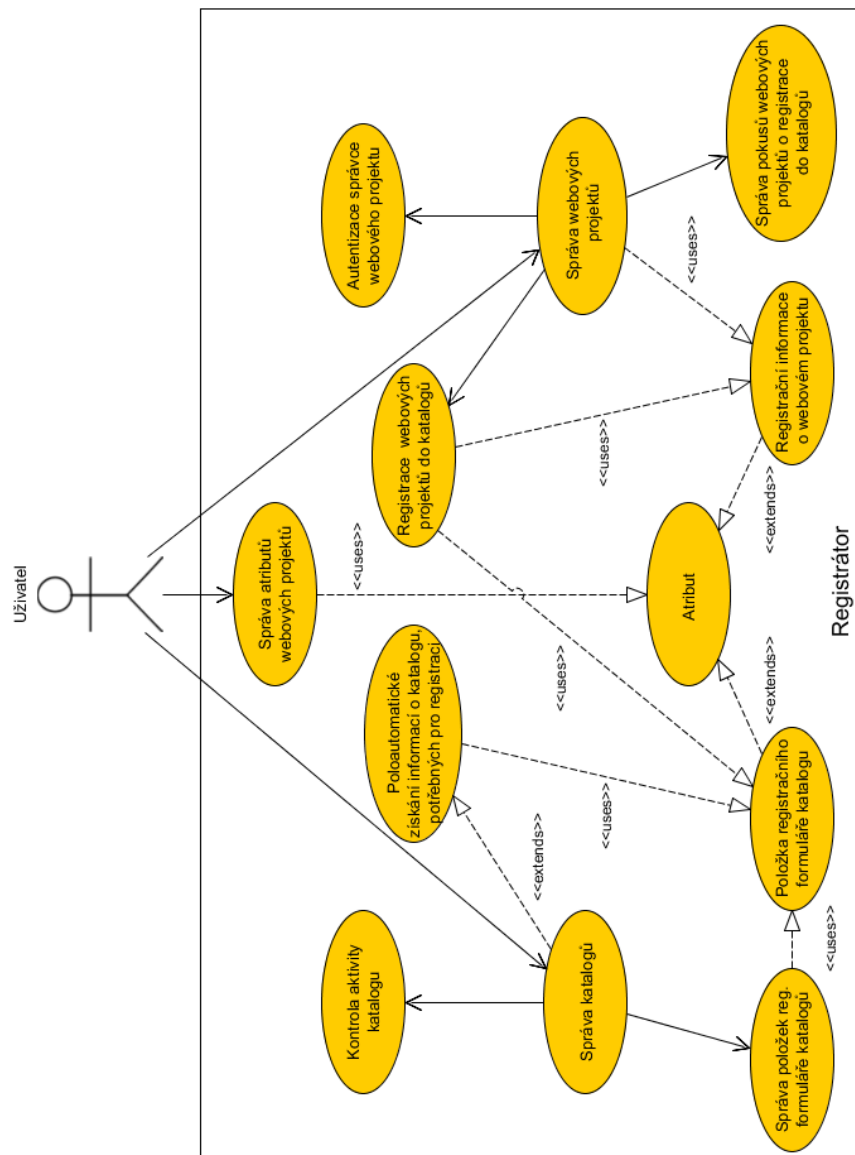
Obrázek 3: Work-flow diagram registrace do katalogu vyžadujícího vytvoření uživatelského účtu



## 5 Analýza informačního systému

Informační systém je soubor lidí, technologických prostředků a metod, které zabezpečují sběr, přenos, zpracování a uchování dat za účelem tvorby prezentace informací pro potřeby uživatelů. Informacemi míníme sdělení, které odstraňuje nejistotu nebo nevědomost, daty míníme jakékoli zaznamenané poznatky či fakta. Jako zvláštní pojem zde vystupuje také znalost představující zobecnění poznání určité části reality. Informaci je možno také chápat jako data s nějakým přidaným významem (data + význam). [15]

Před začátkem vývoje IS musí být nadefinovány požadované funkce, jejich případy užití a typy uživatelů, kterým je povoleno jednotlivé funkce IS využívat. K zachycení různých případů užití IS slouží tzv. use-case diagram (obrázek 4).



Obrázek 4: Use-case diagram Registrátoru

## **5.1 Specifikace požadavků na IS**

Informační systém budou využívat uživatelé, kteří mají přístup k administraci webových projektů, které budou moci zadat do systému a následně je pomocí Registrátoru registrovat do katalogů.

### **5.1.1 Funkční požadavky**

- Přehled a správa atributů webových projektů, požadovaných při registraci do katalogů
- Přehled a správa katalogů
- Ověřování aktivity katalogu
- Sledování kvality katalogů z hlediska (page rank, srnk, alexa) a upozorňovat na případné změny
- Přehled a správa webových projektů
- Autentizace správce webového projektu
- Výběr typů katalogů dle požadavků pro konkrétní webový projekt a automatická nebo polo-automatická registrace webového projektu do katalogů a vyhledávačů na internetu
- Ověření výskytu odkazu na webový projekt v katalogu
- Sledování síly webu vzhledem k registraci do katalogů

### **5.1.2 Nefunkční požadavky**

Informační systém je implementován jako multiplatformní – tedy nezávislý na operačním systému. Díky využití open-source nebo bezplatných nástrojů, framewroků a knihoven není nutné vynakládat velké finanční prostředky na nákup licencí. IS bude vykazovat zvýšené nároky na konektivitu, z důvodu stahování stránek katalogů.

## **5.2 Vstupy IS**

- Atributy webových projektů, požadované při registraci do katalogů
- Obecné informace o katalogích typu webová adresa, jazyk, typ apod.
- Informace o katalogích pro poloautomatické v podobě iMacra pro získání registračních informací o katalogu, nebo tato data vložena manuálně
- Obecné informace o webových projektech typu webová adresa, název, jazyk apod.
- Informace o webových projektech, které budou použity jako údaje při registraci do katalogů

## **5.3 Výstupy IS**

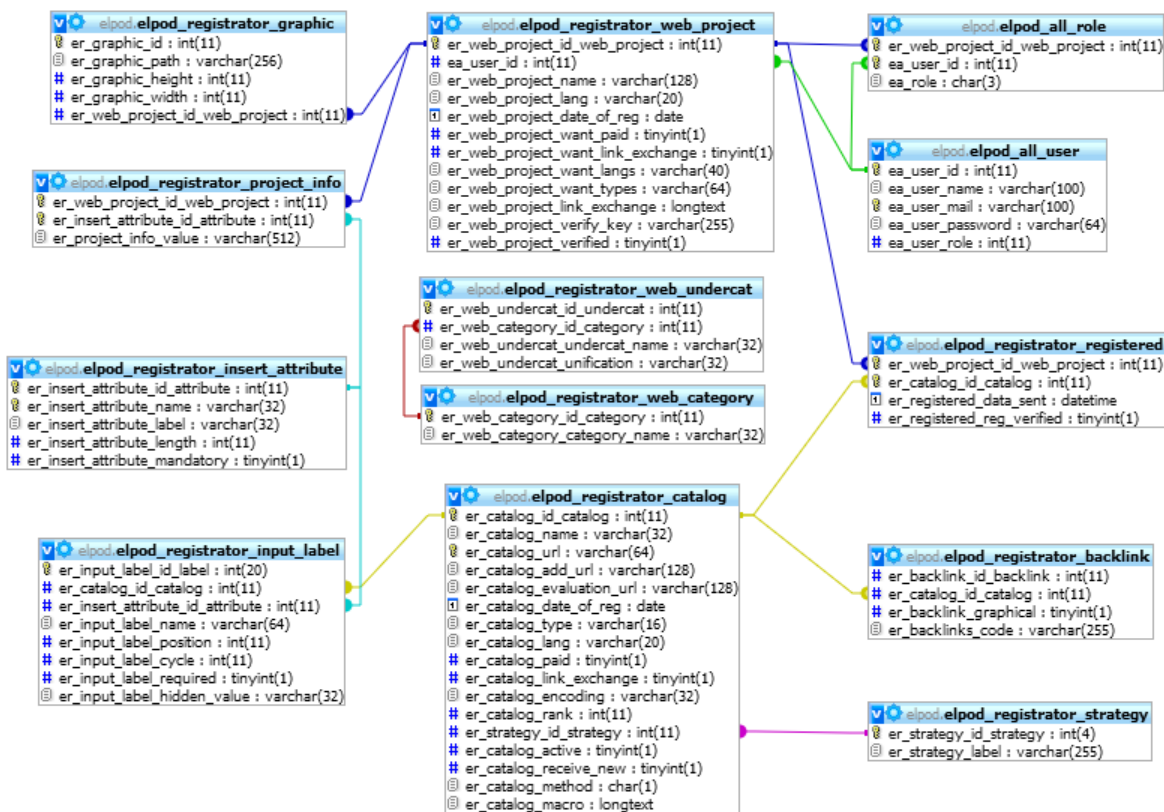
- Přehled atributů webových projektů, požadovaných při registraci do katalogů
- Přehled katalogů a vyhledávačů
- Přehled kvality katalogů v čase

- Přehled položek registračního formuláře katalogu
- Přehled webových projektů
- Přehled informací o webových projektech, které budou použity jako údaje při registraci do katalogů
- Přehled vývoje síly webu vzhledem k registraci v katalogích
- Výsledek automatické registrace do katalogu
- Částečně předvyplněný registrační formulář
- Soubor iMacra
- Přehled pokusů o registraci webového projektu do katalogů

## 5.4 Datová analýza

Databáze patří mezi nepostradatelné komponenty systému. Uchovávají se v ní všechna perzistentní data a struktura jejich uspořádání musí odrážet vazby mezi objekty (entitami) v realitě. Návrh její struktury má vliv na odezvu systému. Proto byla provedena datová analýza, z níž získáme vhodně navrženou strukturu ve 3NF, odrážející specifikace zadání a především realitu.

Diagram zachycuje strukturu databáze v tabulkách, názvy a typy jejich atributů a také jsou v něm vidět relace mezi daty v tabulkách. Nástroj pro správu databáze phpMyAdmin, který je součástí instalace balíku EasyPHP, vygeneroval ER diagram (obrázek 7). [12]



Obrázek 5: ERD modelu databáze

## 6 Implementace webové aplikace

Před započítím vlastní implementace IS bylo nutné zvolit programovací jazyky, run-time prostředí, aplikační rámce (frameworky), knihovny, vývojové nástroje apod. Následující kapitoly toto popisují a uvádějí důvody volby.

### 6.1 Run-time prostředí, testování

Jelikož se jedná o aplikaci, která působí v internetovém prostředí a také se skrze internet k ní přistupuje, zvolil jsem pro zpracování tohoto nástroje skriptovací jazyk PHP. Další pozitivum tohoto jazyka pro použití na tomto projektu spočívají v aktuálnosti jazyka (není zastaralý a stále vznikají nové aplikace), existenci mnoha knihoven a také frameworků, zajišťujících sofistikované systémy zabezpečení proti útokům, efektivní perzistentní vrstvu a ladicí nástroje. [3]

Testování celého systému probíhalo na serverech z balíku EasyPHP, který v sobě kombinuje Apache server, což je běžně používaný webový server, PHP server pro vykonávání příkazů jazyka PHP a MySQL server jakožto server pro perzistentní vrstvu. V balíku EasyPHP jsem viděl výhodu v podobě předkonfigurovaných instancí všech serverů potřebných pro vývoj a chod Registrátoru. [12]

### 6.2 Zvolený SŘBD

Pro účely vývoje využívá Registrátor MySQL server. Jelikož MySQL server patří mezi servery v balíku EasyPHP je tedy logickou volbou. Není ovšem vyloučeno, že by se v budoucnu přešlo k jinému systému řízení báze dat, např. MS SQL či Oracle, které poskytují oproti MySQL pokročilejší datové struktury, což by při rostoucím objemu uchovávaných dat dalo potenciál ke zvýšení propustnosti databázových operací a tím i k rychlejší odezvě systému.

### 6.3 Použité frameworky

Použité frameworky jsou moderní (využívají v současnosti často využívané technologie), těší se značné oblibě nejen u českých vývojářů a díky velké skupině příznivců se můžeme domnívat, že neupadnou v nejbližší době v zapomnění.

#### 6.3.1 Framework Nette

Zde použitým PHP frameworkem je Nette s licencí open-source od českých tvůrců se SEO, reflexí vlastního kódu, podporou moderních technologií jako je např. AJAX a který ctí návrhový vzor MVC (přesněji MVP – Model-View-Presenter). Výhodou je také umístění mezi nejrychlejšími PHP frameworky v nezávislém testu. Nabízí pluginy, rozšíření a ladicí nástroje. Zároveň poskytuje ochranu proti hackerským útokům, jako jsou PHP injection, Cross-Site Scripting, session hijacking, session stealing a další. [8]

#### 6.3.2 Databázový framework dibi

K frameworku pro aplikační vrstvu byl použit databázový framework dibi od stejných tvůrců jako Nette. Framework dibi není vázán pouze na jeden typ SŘBD (podporuje MySQL, PostgreSQL, SQLite, MS SQL, Oracle, Access), čímž se projekt stane přenositelným mezi podporovanými

SŘBD bez složitých úprav. [10]

## **6.4 Knihovny PHP tříd a funkcí**

### **6.4.1 Gridito**

Jedná se o knihovnu pro framework Nette pro dynamické vytváření tabulek s daty s možností mimo jiné pro řazení dle hodnot jednotlivých sloupců a stránkování rozsáhlých seznamů. [9]

### **6.4.2 Simple HTML DOM Parser**

DOM parser je pro použití na stromovou strukturu HTML značek velice užitečný nástroj. DOM parser SimpleHTMLDom byl vybrán pro jeho jednoduchost a všestrannost použití. [11]

## **6.5 Vývojové nástroje**

Editaci PHP lze provádět prakticky v jakémkoliv textovém editoru. Např. editor NetBeans [13] má oproti základním editorům propracovaný systém nápovědy a kontroly syntaxe, takže urychluje práci s objektově orientovanými jazyky. Za základní stavební kámen návrhu databáze lze považovat ER diagram. Ten vznikl v návrhovém softwaru společnosti Quest Software s názvem Toad Data Modeler [14].

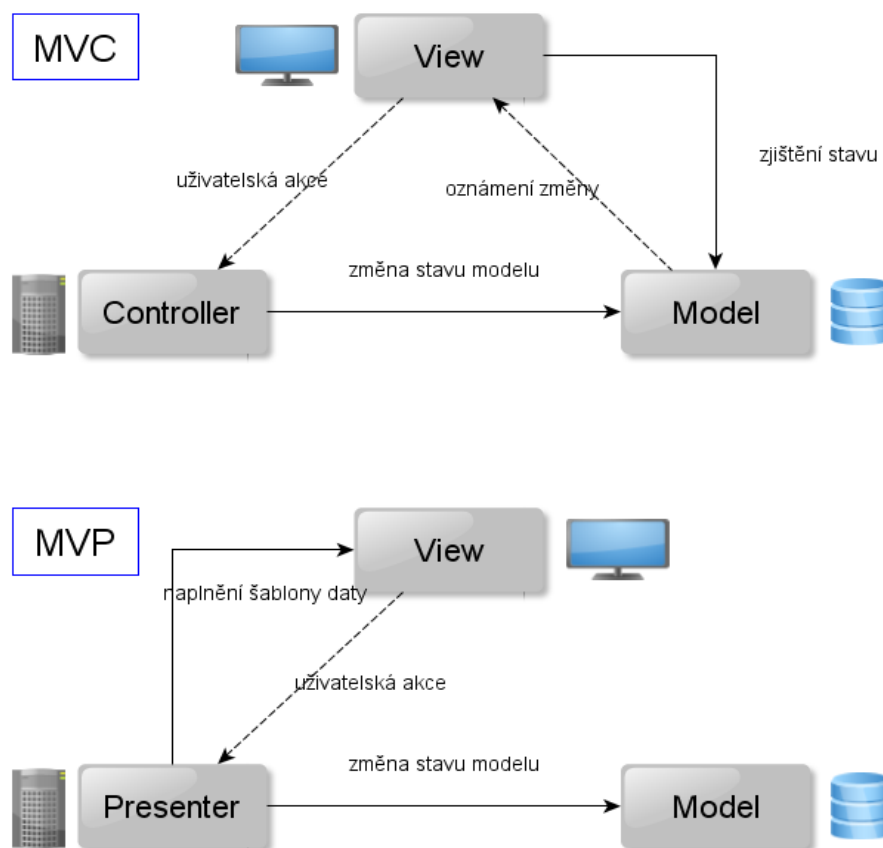
## 7 Struktura

Registrátor, jakožto webová aplikace, lze z několika pohledů rozdělit na části, které mají mezi sebou určité vazby. Jedná se o rozdělení komponent systému dle určitého návrhového vzoru a rozdělení GUI na jednotlivé obrazovky (neboli View) s různými přehledy dat a různými akcemi, které může uživatel provést.

### 7.1 Struktura systému

Framework Nette vychází z návrhového vzoru MVC, resp. MVP, protože v tomto frameworku se Controller nazývá Presenter. Vzor MVP se nejčastěji využívá u komponentových systémů. Datová (Model), řídicí (Controller / Presenter) a prezentační (View) vrstva je oddělená – jednotlivé vrstvy jsou umístěny v samostatných třídách, třídy v samostatných souborech a soubory jednotlivých vrstev v jednotlivých adresářích. Adresářová struktura, použitá u Registrátoru je specifická pro framework Nette [8].

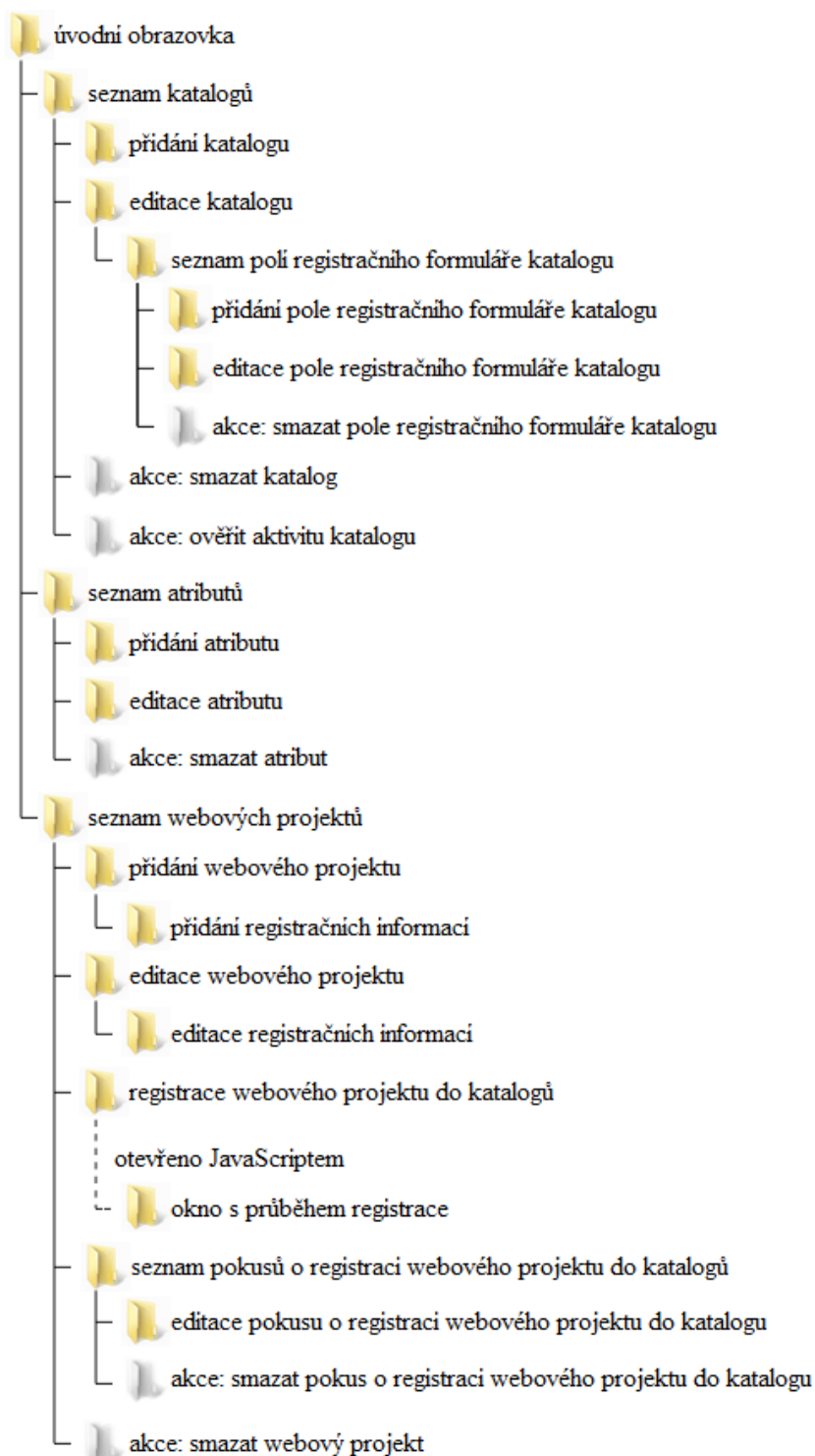
U frameworku Nette je jistá odlišnost v přístupu prezentační vrstvy k datům (obrázek 5). U klasického MVC prezentační vrstva sleduje změny v datech (v Modelu) – při příchodu události o změně dat překreslí objekty, které závisejí na změnách dat. Naproti tomu u vzoru MVP zasílá data prezentační vrstvě Presenter – Controller.



Obrázek 6: Srovnání návrhových vzorů MVC a MVP z hlediska interakce modelu s ostatními vrstvami

## 7.2 Struktura GUI

Hierarchie přístupu k jednotlivým pohledům a akcím, které může uživatel vyvolávat z uživatelského rozhraní, lze popsat stromovou strukturou (obrázek 7).



Obrázek 7: Struktura obrazovek GUI Registrátoru

## 8 Přístup řídicí vrstvy k perzistentní vrstvě a plnění databáze daty

Připojení řídicí vrstvy k perzistentní obstarává framework dibi skrz jeho MySQL ovladač. Takových ovladačů dibi obsahuje ještě několik pro připojení k dalším z podporovaných SRBD.

### 8.1 ORM

Úlohu objektově-relačního mapování opět zvládá framework dibi. [10] Přístup k datům skrz objekty je odlišný od dalších databázových frameworků jako např. Doctrine, nebo Hibernate (pro Javu), kdy bychom získali data v instancích entitních tříd. Dibi se snaží být univerzálnější z hlediska jednoduchosti použití výsledných objektů a nabídnout vyšší pohodlí při programování (různé návratové objekty – dle libosti programátora) za cenu nedodržování striktní podoby objektů s daty. Tento způsob implementace odráží myšlenku tzv. KISS, tedy „Keep It Simple, Stupid“, což nepřímě znamená „méně kódu = méně chyb“.

Struktura a typ návratových hodnot vykonaného dotazu může být:

- instance či pole instancí třídy DibiRow [10], jejíž proměnné mají shodné názvy jako názvy sloupců projekce dle SQL dotazu
- pole instancí třídy DibiRow [10], kde klíč pole je hodnota některého ze sloupců projekce dle SQL dotazu
- pole hodnot (jediný sloupec)
- jediný výsledek – použití pro např. agregační funkce: count(), avg() atd.

Na obdobném principu lze záznamy vkládat do DB nebo je upravovat. Do asociativního pole stačí vložit upravované údaje, kde klíčem je název sloupce a hodnota pak aktuální data, pro uložení do DB. Toto pole poté předat funkci pro vykonání dotazu „query()“. [10]

### 8.2 Plnění databáze daty o katalogích a vyhledávačích

Po vytvoření tabulek v databázi byly tyto samozřejmě zcela bez dat. K jejímu naplnění jsem vytvořil dva PHP parsery. Oba parsery převádějí data ze vstupu na příkazy INSERT pro databázi. Namísto přímého vkládání do DB jsem zvolil tento způsob z důvodu přenositelnosti výsledného skriptu a možnosti jeho opakovaného použití. Protože jsou tyto parsery určeny pouze pro jednorázové použití, nejsou zakomponovány do systému Nette.

#### 8.2.1 Parser vstupních dat na internetu

Tento parser procházel seznam katalogů a vyhledávačů, umístěný na serveru [www.seznamkatalogu.cz](http://www.seznamkatalogu.cz) [7]. Využíval přitom stránkování katalogů (postupný výpis po cca 50ti katalogích) namísto načtení všech najednou. Důvod k tomuto rozhodnutí byl v použití DOM parseru SimpleHtmlDom [11]. DOM parsery potřebují pro svou činnost značnou kapacitu operační paměti, která není neomezená. Při počtu přes 2000 katalogů v HTML tabulce by mohlo při zpracovávání skriptu (parseru) dojít buď k zamrznutí počítače, či ke zpomalení jeho odezvy a celkové rychlosti zpracování dat.

Umístění souboru parseru: „\parsery.zip\parser.php“.



## 8.2.2 Parser vstupních dat v souborech CSV

Data od vedoucího této bakalářské práce pana Ing. Radoslava Fasugy PhD., byla původně v souborech formátu XSLX, ale pro snazší práci s daty v souborech jsem tyto převedl do souborů CSV.

Tento parser využívá jednoduchosti CSV souboru a funkcí explode() rozděluje do pole jednotlivé hodnoty, se kterými se pak bezproblémově pracuje. Parser před přidáním katalogu kontroluje, zda se aktuálně zpracovávaná webová adresa již v databázi nenachází, aby nevznikaly duplicity.

Umístění souboru parseru: „\parsery.zip\parser\_csv.php“.

## 8.3 Přidávání dalších katalogů a vyhledávačů - parser maker iMacros

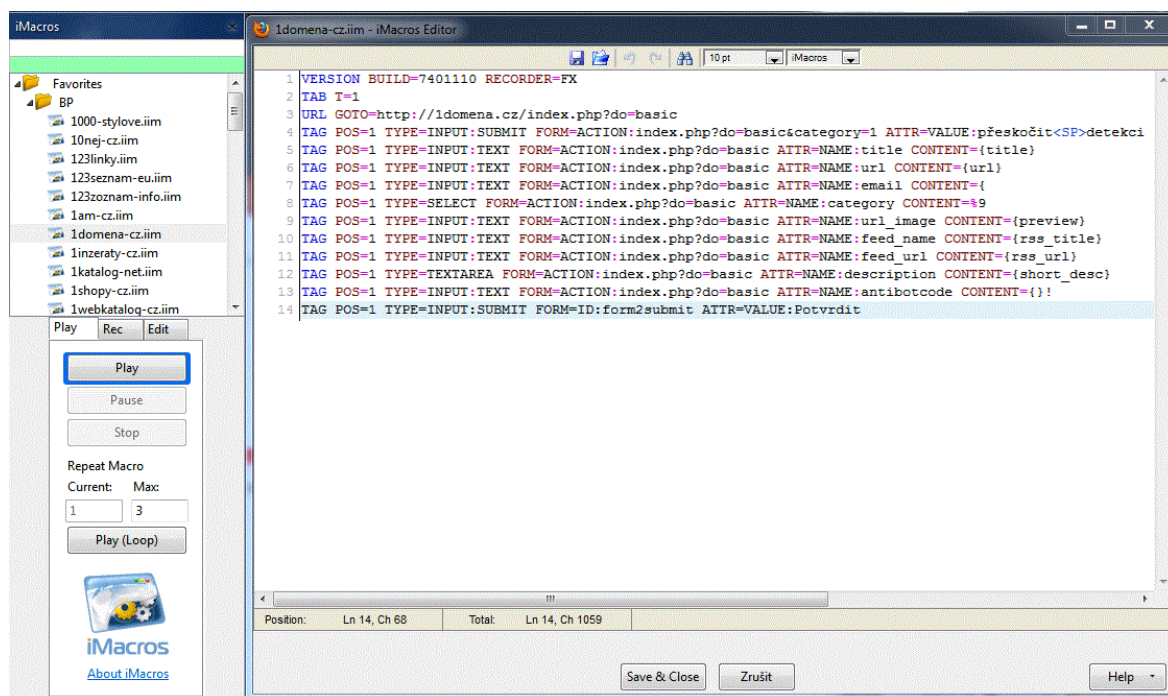
Kromě základních údajů, jako je např. URL adresa katalogu, jazyk, typ a údaje zda se jedná o placený katalog, nebo katalog vyžadující zpětný odkaz, je potřeba také dodat informace, které poslouží jako nástin struktury registračního procesu. Registrátor by jinak nevěděl, kam jaká data poslat, nebo co kam vyplnit. Buď lze tato data zadat ručně, tak jak byly zadávány základní údaje, nebo i tento postup alespoň částečně zautomatizovat. Poloautomatizované shromažďování informací o katalogu se provádí záznamem postupu, který by jinak musel uživatel manuálně provést, aby dosáhl zaregistrování webového projektu do katalogu. Záznam činností (makro) je znám také například z tabulkového procesoru MS Excel. V našem případě se o záznam postará zásuvný modul pro webový prohlížeč.

iMacros je zásuvný modul pro internetový prohlížeč, existující ve verzi pro Internet Explorer, Mozilla Firefox a Google Chrome. Tento zásuvný modul zaznamenává akce uživatele (URL adresa, kliknutí na stránce, zadávání hodnot z klávesnice atd.), která lze „přehrát“ a tím je opětovně vykonat. Výsledné makro je textový soubor s příponou „iim“, obsahující na jednotlivých řádcích direktivu ohledně provedené akce uživatele. Pro naše použití se bude jednat o příkazy pro přechod na určitou URL adresu a příkazy, týkající se formulářových prvků. V makru se tudíž budou nacházet příkazy pro vyplňování formulářových polí, výběru kategorií a případně také potvrzení formuláře. [6]

Po zaznamenání makra jej musíme ještě předat Registrátoru. Lze to provést při vytváření nového katalogu, nebo při editaci již uloženého. Registrátor přijímá iMacro buď jako soubor, nebo jako prostý text, který soubor s příponou „iim“ obsahuje. Nově načtené, nebo upravené makro je předáno parseru, skrytého ve funkci s názvem „GatherIMacro()“. Parser se nachází v knihovně funkcí „\REGISTRATOR.zip\libs\Ext\Fcs.php“.

Nutno podotknout, že plugin iMacros pro webový prohlížeč Google Chrome nefungoval v době vývoje Registrátoru zcela korektně při záznamu makra. Vytvořená makra neobsahovala všechna data důležitá pro parser Registrátoru. Z toho důvodu bylo nutné využívat pro záznam maker plug-in iMacros ve webovém prohlížeči Mozilla FireFox.

Ukázka GUI pluginu iMacros pro Mozilla Firefox – vlevo panel pro výběr a ovládání (záznam, přehrání) makra, vpravo editační okno pro úpravu jednoho z maker (obrázek 8).



Obrázek 8: GUI plug-inu iMacros pro Mozillu Firefox

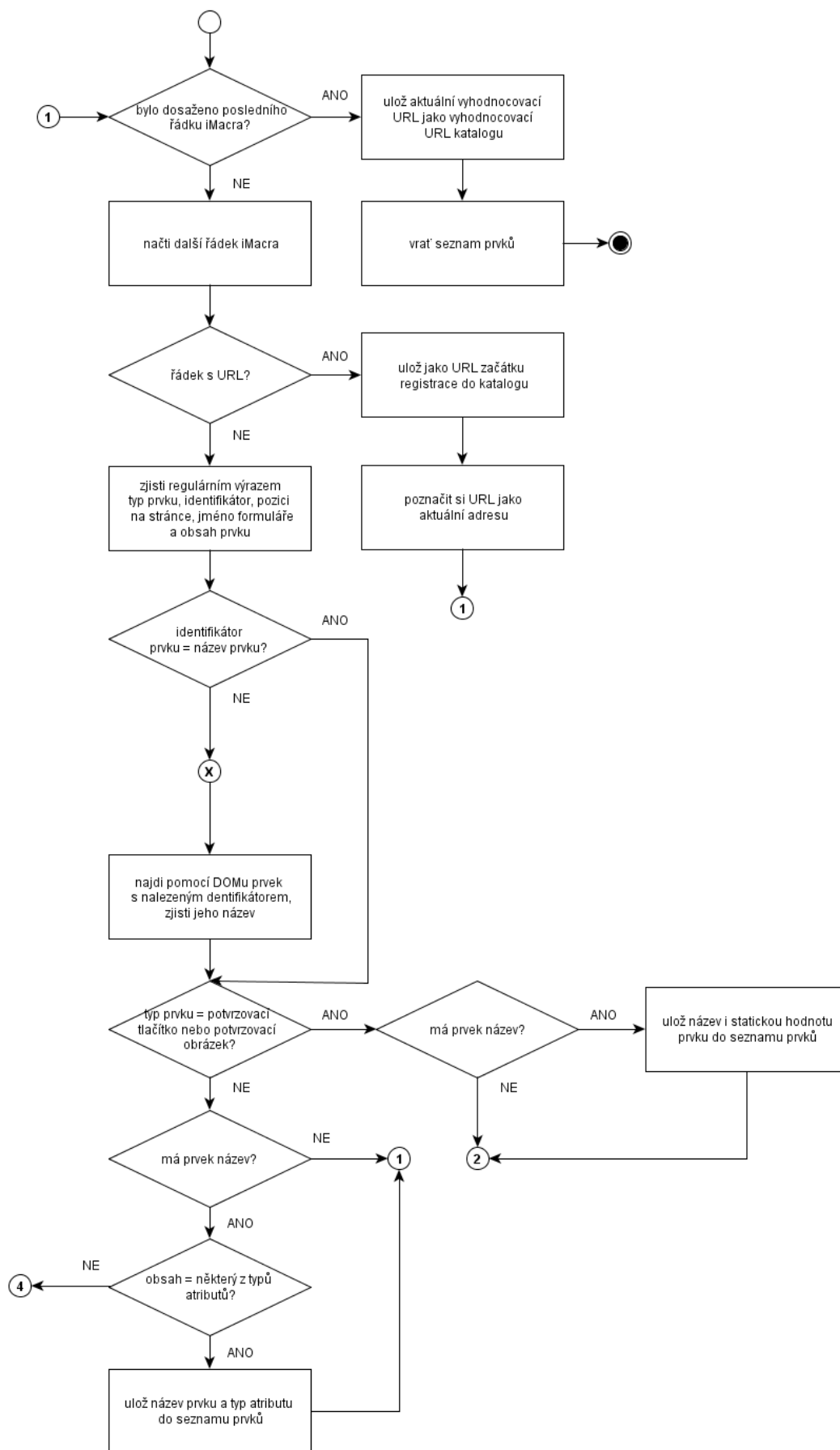
### 8.3.1 Co vše dokáže parser z maker vytěžit

- Vstupní formulářové prvky, jejichž seznam je prakticky uveden v iMacru – nicméně je nutné vždy získat název pole, ale iMacro někdy obsahuje ID formulářového prvku
- Který údaj do daného políčka patří (zde je nutné uzavírat ID nebo název atributu, odpovídající databázi, do složených závorek)
- Skrytá políčka, přepínače (radio button), zaškrtnutí políčka (i která se v iMacru nevyskytovala), potvrzovací tlačítka
- Výše uvedené údaje dokáže získat ze všech URL stránek, kterými je potřeba projít pro úspěšné provedení registrace a zjišťuje také, ve kterém kroku registrace se formulářový prvek nachází
- URL pro přidání odkazu (začátek registrace) a pro vyhodnocení formuláře
- Metoda odesílání dat formuláře (GET / POST)

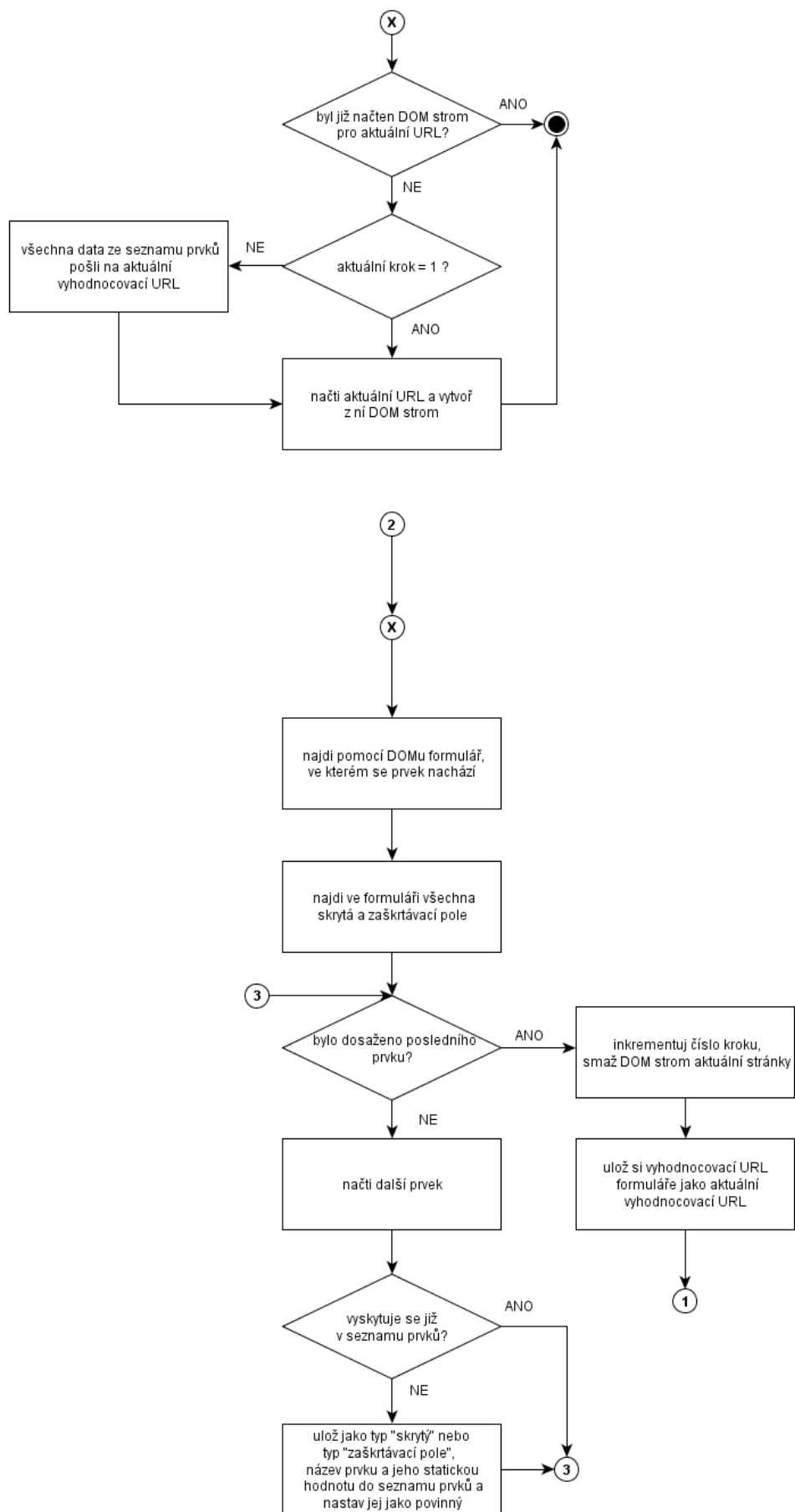
### 8.3.2 Princip činnosti parseru

Parser při své činnosti využíval regulární výrazy [5], nativní funkce PHP jazyka pro práci s řetězci a DOM parser [11].

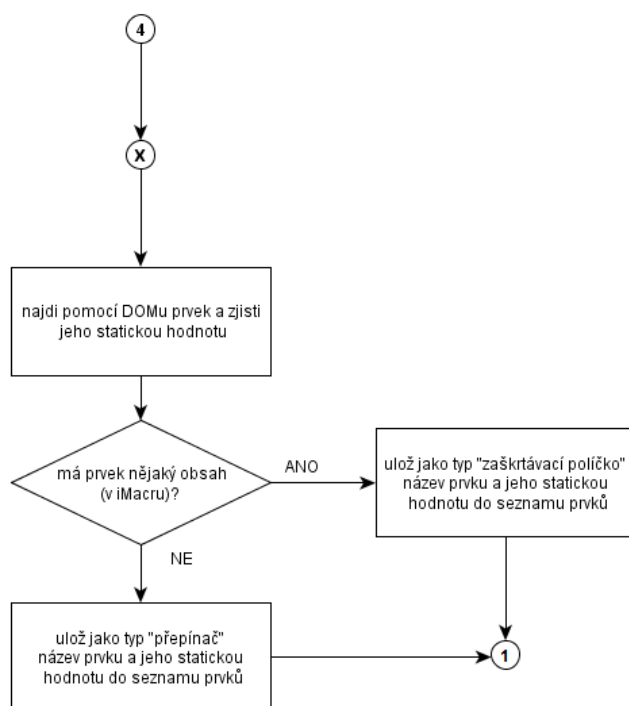
Vývojový diagram (obrázky 9- 11) znázorňuje postup, jakým parser získává data (viz. předchozí kapitola). Čísly v kruhu jsou označeny odkazy na další části diagramu, písmenem v kruhu je označen podprogram, po jehož ukončení se bude pokračovat v místě, odkud byl zavolán.



Obrázek 9: Vývojový diagram parseru maker, část 1



Obrázek 10: Vývojový diagram parseru maker, část 2



**Obrázek 11: Vývojový diagram parseru maker, část 3**

Ke každému z prvků se ukládá pořadové číslo kroku registrace a pořadí z prvků se stejným jménem ve struktuře formuláře (identický název v jediném formuláři mají obvykle zaškrtnutí políčka).

## 9 Implementace funkcí IS

Následující kapitoly popisují funkci a popis činnosti metod, které zpracovávají data, týkající se webových HTML stránek (od katalogů nebo webových projektů). Zpracované informace poskytují aplikaci k dalšímu zpracování či rozhodování, nebo uživateli pro poskytnutí důležitých informací nebo určité funkcionality systému.

### 9.1 Zjištění použité znakové sady

Interní funkce pro správnou funkci některých strategií Registrátoru. Tato informace je důležitá při posílání hodnot skriptu, který vyhodnocuje data z formuláře, nebo při předvyplňování formulářových polí. Pokud cílový katalog používá jinou znakovou sadu než UTF-8, jsou data o webovém projektu před jejich použitím transformována funkcí `iconv()`.

Parser, který získává informaci o znakové sadě, ji najde pomocí regulárního výrazu v některém z meta tagů zdrojového kódu stránky.

### 9.2 Převod relativních adres na absolutní

Interní funkce pro správnou funkci některých strategií Registrátoru. Díky této funkci se například stránky s předvyplněným formulářem zobrazují a chovají identicky, jako kdyby byly ke klientovi načteny přímo z dané domény, namísto z Registrátoru. Funguje tak především registrační formulář, ale také všechny hypertextové odkazy, URL adresy, směřující na externí JavaScripty, CSS soubory či obrázky.

Princip fungování spočívá v rozparsování zdrojového kódu stránky regulárním výrazem a získání všech URL adres. Tím je myšleno všechny atributy „href“, „src“ a „action“. Následně je pole se získanými adresami v cyklu procházeno a u každého adresy se zjišťuje, zda je relativní. Pokud je relativní, musí se ještě rozhodnout, co do adresy přidat, aby fungovala správně – relativní adresy mohou odkazovat od aktuálního umístění nebo od kořene domény. Upravená URL je přidána do pole s prvky, kterými bude nahrazováno, původní URL do pole s prvky, které budou nahrazeny. Do těchto polí musejí být předávány textové řetězce včetně HTML atributu („href“, „src“ či „action“). Poté funkce `str_replace` provede hromadné nahrazení řetězců.

### 9.3 Detekce použití enginu pro katalogy

Tak jak existuje například engine pro fóra phpBB3, stejně tak existují enginy pro vytváření katalogů stránek. Výhoda použití enginu spočívá v získání hotového odladěného řešení, které je potřeba pouze graficky upravit a lokalizovat texty. Právě díky tomuto lze pozorovat jistou podobnost zdrojových HTML kódů webů.

Skript tedy porovnává zdrojový kód katalogu s referenčními katalogy, u který bylo předem manuálně ověřeno, že daný engine používají. Dále zjišťuje přítomnost fragmentu – tedy kusu kódu, obvykle obsahující část HTML odkazu, vedoucí na autora enginu. Dále má skript nastaven 2 limity – horní a dolní limit. Pokud procentuální podobnost mezi zdrojovými kódy referenčního a porovnávaného katalogu přesáhne horní limit, je tento katalog považován jako využívající daný engine a podobnost s následujícími typy enginů se již netestuje. V ostatních případech se vždy porovnává zdrojový kód porovnávaného katalogu se zdrojovými kódy všech referenčních katalogů. Pokud nejvyšší procentuální podobnost zdrojových kódů katalogu s některým ze

zdrojových kódů referenčních katalogů přesahuje dolní limit a zároveň obsahuje fragment, je opět považován jako využívající daný engine. Bez výskytu fragmentu se pouze objeví zvýraznění v tabulkovém výpisu u tohoto katalogu a enginu – jako ukazatel pro manuální kontrolu. Tato funkce vrací text, formátovaný pomocí HTML značek, který tvoří výpis výsledků porovnávání (obrázek 12).

ID	katalog	<a href="http://www.sefdir.org">http://www.sefdir.org</a> (eSyndicat)	frag.	<a href="http://links.eu">http://links.eu</a> (Free-Globes)	frag.	<a href="http://links.hys.cz">http://links.hys.cz</a> (Free-Globes - starší verze)	frag.	<a href="http://categorizedirectory.com">http://categorizedirectory.com</a> (phpLD)	frag.	<a href="http://1000webu.com">http://1000webu.com</a> (Link Up)	frag.
1643	<a href="http://workline.cz/katalog.aspx">http://workline.cz/katalog.aspx</a>	6.2380536462487%		7.35232668566%		7.5768450616321%		3.7458871171855%		4.2501517911354%	
1644	<a href="http://worldfamousdirectory.org">http://worldfamousdirectory.org</a>										
1645	<a href="http://worldindexsite.com">http://worldindexsite.com</a>										
1646	<a href="http://worldwebmaster.tk/">http://worldwebmaster.tk/</a>	2.9468549944111%		2.6261106334193%		4.3978653635954%		8.8346733203955%		8.6342123056119%	
1647	<a href="http://wpkatalog.mzf.cz/">http://wpkatalog.mzf.cz/</a>	5.0036637705433%		5.3384279475983%		3.5768096953816%		7.7774569255944%		11.143613316618%	
1649	<a href="http://www.katalog.ic.cz">http://www.katalog.ic.cz</a>	3.538314415201%		46.624464067309%	x						
1650	<a href="http://wvz.cz/">http://wvz.cz/</a>	3.3531222515391%		1.1147821005551%		2.439990942029%		8.6315585297255%		6.0386870513453%	
1651	<a href="http://x-obchody.cz/">http://x-obchody.cz/</a>										
1652	<a href="http://x-portal.cz">http://x-portal.cz</a>	8.2776011402259%		3.0891498102957%		4.9352705070948%		11.466074703214%		7.453959232745%	
1653	<a href="http://xaa.cz/">http://xaa.cz/</a>	3.3531222515391%		1.1147821005551%		2.439990942029%		8.6315585297255%		6.0386870513453%	
1654	<a href="http://xce.cz/">http://xce.cz/</a>										
1686	<a href="http://zapisse.cz">http://zapisse.cz</a>	2.9903065421898%		55.422976342517%							
1687	<a href="http://zapisweb.eu">http://zapisweb.eu</a>	3.4443464196442%		30.256848678586%	x	22.316336799732%	x	3.9031100221377%		9.036962361149%	
1688	<a href="http://zaregistrovatstranky.cz">http://zaregistrovatstranky.cz</a>	3.0652837962634%		59.72230628624%							
1689	<a href="http://zaregistrovatweb.cz">http://zaregistrovatweb.cz</a>	2.9239661435499%		30.733476102404%		31.32061518235%		5.3497770926211%		7.3006792992492%	
1690	<a href="http://zaregistrujodkaz.cz">http://zaregistrujodkaz.cz</a>	2.628287764801%		53.161969491785%	x						
1691	<a href="http://zasilkac.cz">http://zasilkac.cz</a>	2.9761766684451%		6.4060292039567%		5.3447375041121%		1.8688486921889%		4.958562735439%	
1692	<a href="http://zbozi.cz/">http://zbozi.cz/</a>	2.6980316850696%		1.2860316946484%		2.2110347642697%		8.1825986559697%		4.9044753937244%	
1693	<a href="http://zdamadokatalogu.cz">http://zdamadokatalogu.cz</a>	2.9898848700177%		56.778223311893%	x						
1694	<a href="http://zdroj.cz">http://zdroj.cz</a>	4.2261061250197%		1.6260162601626%		2.1885418584198%		12.423557007623%		13.06425248334%	
1695	<a href="http://zdrojinformaci.cz">http://zdrojinformaci.cz</a>	2.548968006316%		53.238405514535%	x						
1696	<a href="http://zebricek.cz">http://zebricek.cz</a>	5.1041704807587%		5.2973214951876%		5.4086890003825%		13.844570252187%		12.657117801812%	
1697	<a href="http://zetakatalog.info">http://zetakatalog.info</a>	3.5233285233285%		46.700001312388%	x						
1698	<a href="http://zhledni-to.cz">http://zhledni-to.cz</a>										

Obrázek 12: Příklad výpisu série porovnávaných katalogů s katalogovými enginy

Upozornění – tento skript má dlouhou dobu vykonávání, která obvykle překročí maximální možnou dobu běhu skriptu na hostinzích (20 – 30 s).

## 9.4 Kontrolování aktivity katalogu či vyhledávače

Význam kontrolování aktivity spočívá v zamezení zbytečného snížení rychlosti registrace při pokusech o registraci na neaktivních katalogích.

Skript se pokouší ze serveru získat hlavičku HTML dokumentu pomocí PHP funkce „get\_headers“. Je-li možné hlavičku získat (server odpovídá) je poté rozložena na jednotlivé typy informací pomocí PHP funkce „explode“. Pokud není možné se serverem navázat spojení (a tím pádem nelze načíst hlavičku), nebo obsahuje-li hlavička HTTP statusový kód větší než číslo 400, pak je katalog označen za neaktivní.

## 9.5 Autentizace majitele (správce) webového projektu

Protože chceme předejít zneužití, například vyplněním hanlivých, zavádějících či nepravdivých údajů, které by se poté registrovaly do katalogů a k jejich změně by měl přístup pouze ten, který je do katalogů registroval, je potřeba ověřit, zda je uživatel opravdu oprávněn provádět takovéto úkony.

Autentizace je jednoduchá – uživatel musí prokázat, že má přístup k administraci webového projektu. Toto provede vložením textového souboru, s Registrátorem vygenerovaným kódem,

do kořenového adresáře svého webového prostoru. Před prvním pokusem o registraci Registrátor kontroluje přítomnost souboru a jeho obsahu. Při splnění je uživateli povoleno zahájit registraci webového projektu Registrátorem.

## **9.6 Registrace webových projektů do katalogů**

Toto je hlavní požadovaná funkce Registrátoru – je proto detailněji popsána v kapitolách 10 a 11.

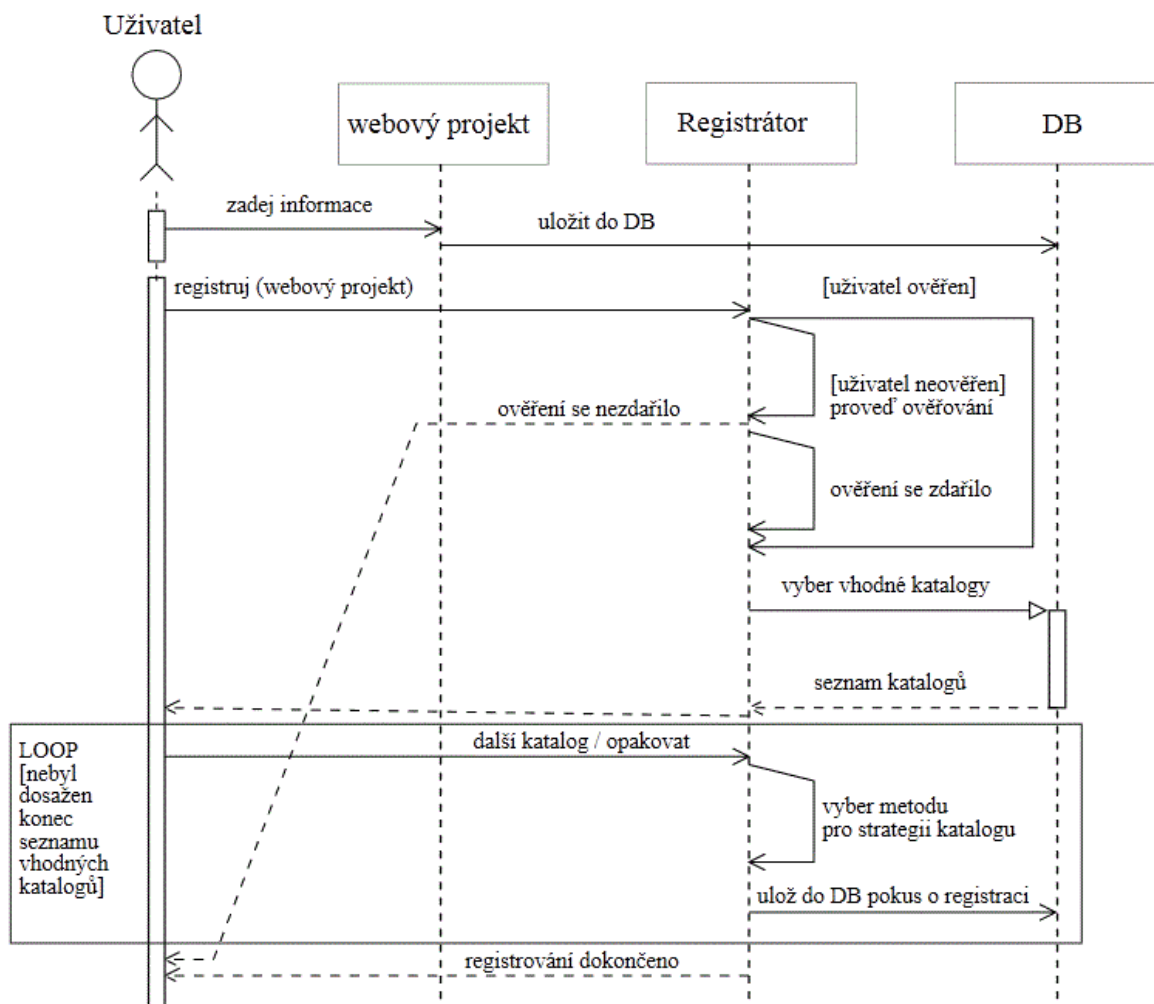
## **9.7 Ověření existence odkazu na webový projekt v katalogu**

Tato funkce nebyla implementována z důvodu nenalezení techniky prohledávání celého webu katalogu, za účelem nalezení odkazu, který směřuje na registrovaný webový projekt v rozumném čase. Navíc byla vypořádována skutečnost, že některé katalogy používají přesměrování. URL odkazu tedy nijak nepřipomíná URL, kam směřuje. Takovéto chování odkazů by automatizované ověřování existence odkazu v katalogu pravděpodobně zcela znemožnilo.



## 10 Proces registrace

Jak bylo výše napsáno, uživateli není povoleno zahájit registraci do katalogů, dokud není ověřeno, že má přístup k administraci projektu nebo pokud nevyplnil všechny povinné údaje o webovém projektu. V případě splnění podmínek dochází k přechodu k řízení registračního procesu, zachyceného na diagramu (obrázek 13).



Obrázek 13: Sekvenční diagram řízení registračního procesu

Aby bylo vůbec co registrovat, musí uživatel nejprve vytvořit webový projekt a zadat mu povinné údaje pro registraci. Seznam všech atributů webového projektu, jejich maximální počet znaků pro každý projekt a zda musejí být vyplněny, než bude povolena registrace, lze procházet v pohledu „Seznam atributů“. Ze stejné obrazovky lze také atributy administrovat (přidávat, upravovat či mazat).

Pokud se uživatel rozhodne, že by již chtěl některý ze svých webových projektů registrovat, stiskne tlačítko „Registrovat!“. Jestli už byl u zvoleného projektu jednou ověřen majitel (správce), je umožněno zahájit registraci. V opačném případě Registrátor provede kontrolu ověřovacího

kódu (viz. kapitola 9.5) a vypíše výsledek ověřování. Není-li ověřování úspěšné, vrátí se Registrátor na seznam webových projektů uživatele.

Operací „vyber vhodné katalogy“ (obrázek 12) se rozumí SQL příkaz, který vyselektuje katalogy, splňující tyto podmínky:

- má definován typ registrační strategie
- je aktivní
- přijímá nové registrace
- poslední pokus o registraci webového projektu byl naposledy nejméně před 60ti dny a zároveň pokus zatím nebyl označen jako úspěšný (nebylo ověřeno, že odkaz na webový projekt se v katalogu opravdu vyskytuje)
- splňuje požadavky uživatele pro registrovaný webový projekt (jazyk a typ katalogu, zda chce také placené a zda chce katalogy, vyžadující zpětný odkaz)

Zde dostává uživatel možnost výběru pomocí tlačítek, zda chce registrovat do dalšího katalogu v pořadí, nebo se chce pokusit o opakovanou registraci do posledního katalogu. Tato možnost se zde nachází pro případy, kdy nebylo možné provést registraci, např. kvůli výpadku spojení ať už mezi uživatelem a Registrátorem, nebo byl problém ve spojení s katalogem.

Funkce, která vybírá obslužné metody pro strategii katalogu, kontroluje reflexí, zda obslužná metoda opravdu existuje. Pokud ano zavolá ji, pokud ne přejde k dalšímu katalogu ze seznamu.

## 11 Registrační strategie

Pod tímto pojmem je označován proces, během kterého provede Registrátor celou registraci nebo její určitou část a případnou další část provede uživatel – lidská obsluha. Strategie se liší v míře nutného zásahu obsluhy a jsou vázány na jistý typ průběhu registrace odkazu do katalogu.

Takových strategií lze vytvořit velké množství. Již teď se u různých katalogů liší a lze předpokládat, že se budou dále vytvářet nové. Proto bylo potřeba vytvořit několik základních, které pokryjí většinové typy registrací a k nim přidat takové, které lze používat univerzálně, ale na úkor toho ne tak moc efektivně. Efektivností je v tomto případě poměr automatizované části registračního procesu ku části, kterou musí vykonat uživatel.

### 11.1 Nepoužité strategie

Následující strategie nebyly implementovány buď z důvodu určitého omezení technického charakteru, nebo z důvodu shledání takové strategie nepřinášející žádný pokrok v registrování do katalogů.

#### 11.1.1 Stránka s registračními daty, iframe uvnitř

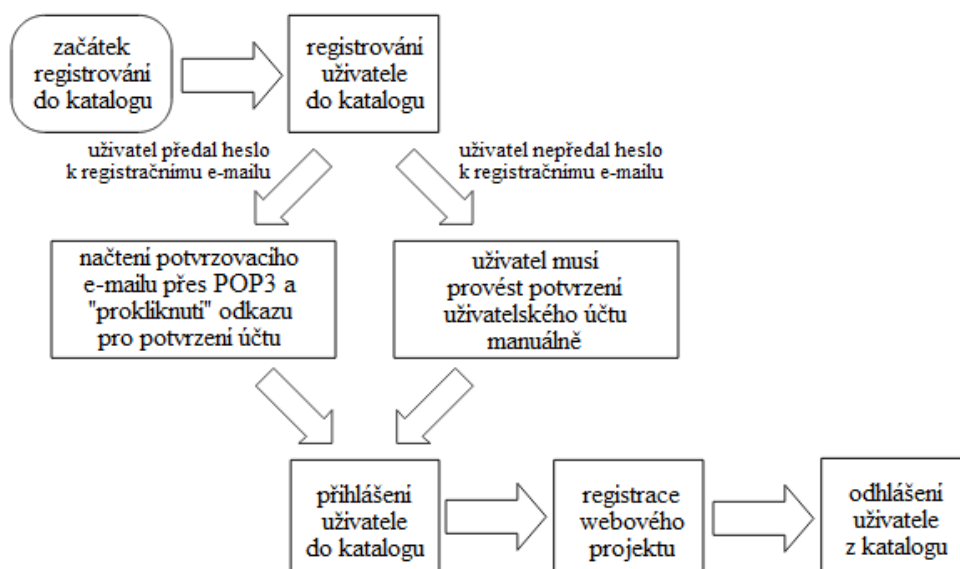
Dalším z možných strategií bylo vytvoření stránky s textovými poli s údaji, potřebnými pro registraci a uvnitř této stránky by se nacházel prvek iframe s načtenou adresou katalogu, kde by probíhala vlastní registrace. Příklad takového zpracování lze vidět na <http://top.clonet.eu/registrace/online-registrace-do-ceskych-katalogu.php>. Bohužel tento způsob nebyl vhodný z několika důvodů. Hlavním důvodem, který by při použití této strategie snižoval uživatelský komfort používání Registrátoru, by byl fakt, že vlastní registraci provádí uživatel, což by při počtu katalogů v řádu stovek až tisíců, bylo velice únavné a navíc by se nedalo hovořit o zautomatizování procesu registrace. Dále omezená podpora prohlížeči - pouze Internet Explorer implementuje možnost kopírování dat do systémové schránky, řízenou JavaScriptem. Uživatel by tudíž musel buď data z textových polí opakovaně ručně kopírovat, nebo používat prohlížeč Internet Explorer (tím pádem zde vzniká omezení na operační systém – Windows i omezení na webový prohlížeč) a poté by měl možnost využít tlačítka, která by po stisknutí zkopírovala daná data do schránky. Takto snížený uživatelský komfort by vedl k nízké oblibě Registrátoru.

#### 11.1.2 Předvyplňování polí formuláře pomocí JavaScriptu

Předvyplňování polí s využitím JavaScriptu mělo fungovat pro vnitřní stránku, načtenou v IFRAMu (vnořené okno). Bohužel toto nelze provést kvůli zabezpečení, implementovanému v prohlížečích – nelze JavaScriptem zasahovat do webové podstránky, pocházející z jiné domény, než na jaké je umístěna hostitelská webová stránka.

#### 11.1.3 Vyžadováno vytvoření uživatelského účtu pro registraci

Některé katalogy vyžadují, aby si uživatel vytvořil uživatelský účet. Až po potvrzení ověřovacím odkazem, který přijde uživateli v e-mailu do e-mailové schránky, je uživateli umožněno registrovat odkazy na webové projekty. Plán činností, které je nutné provést pro úspěšné registrování lze zakreslit do work-flow diagramu (obrázek 14).



**Obrázek 14: Work-flow diagram registrace do katalogu, vyžadující vytvoření uživatelského účtu**

Toto je pravděpodobně nejsložitější situace, kde musí stroj (či lidská obsluha) projít přes 3 různé formuláře – registrační formulář nového uživatele, přihlašovací formulář a formulář pro registraci webového projektu, ve kterém se opět mohou skrývat různé kombinace odlišností, popsanych v kapitole 4.3 a podkapitolách.

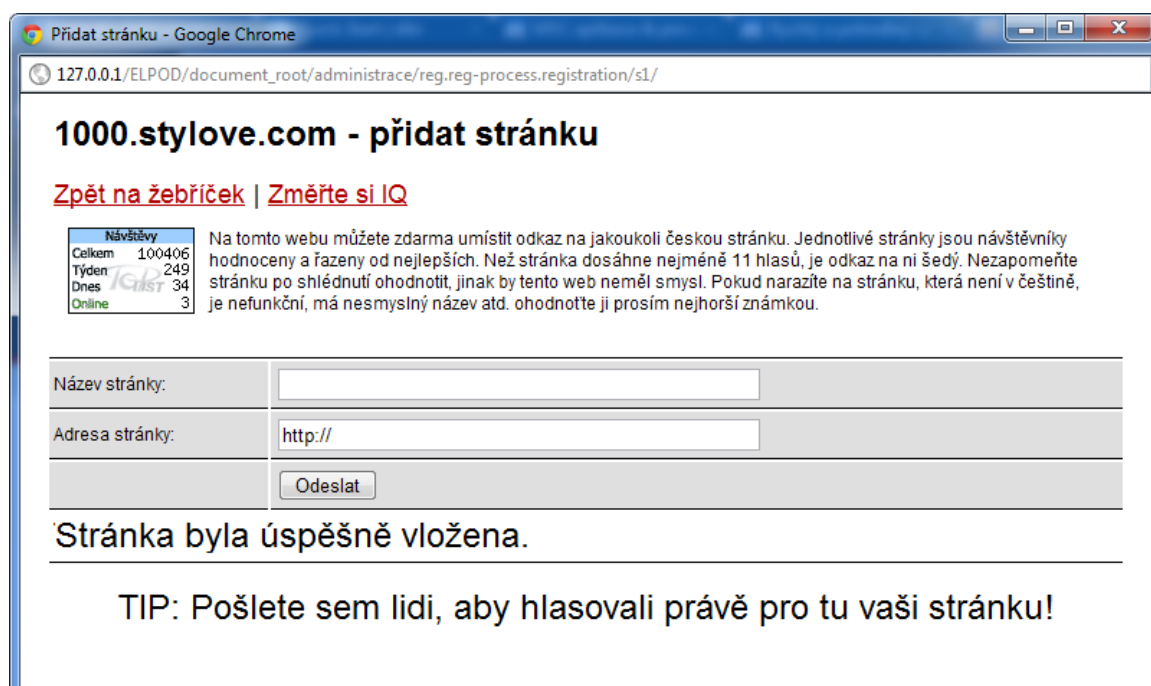
Složitost této strategie a shledání jejího nemalého významu až v pozdní fázi vývoje systému znemožnily její praktickou implementaci.

## 11.2 Všeobecné typy strategií

### 11.2.1 Základní typ – jednoduchý registrační formulář

Pro registraci do katalogu s touto strategií je nutné pouze vyplnit jeden formulář informacemi o webovém projektu.

Data z formuláře jsou poté odeslána skriptu pro zpracování dat z tohoto formuláře, uvedeném v atributu „action“ formuláře ve tvaru: `název_pole1=hodnota_pole1&název_pole2=hodnota_pole2&....`. Této skutečnosti lze využít a posílat na adresu vyhodnocovacího skriptu data přímo a to buď metodou GET či POST (dle nastavení registračního formuláře daného katalogu). Před odesláním dat musí proběhnout jejich konverze do znakové sady katalogu. Po registraci do katalogu se uživateli pouze zobrazí výsledek registrace – zda byl odkaz přidán, nastala nějaká chyba, odkaz už v katalogu existuje apod. (obrázek 15).



Obrázek 15: Příklad výsledku registrace

Před výpisem stránky (metodou `print()`) je nutné změnit v HTTP odpovědi (HTTP response) kódování na takové, které používá katalog. V opačném případě by webový prohlížeč vykreslil místo českých a jiných speciálních znaků pouze čtverečky a podobné nic neříkající symboly a teprve ruční změna kódování v nastavení webového prohlížeče by toto mohla napravit.

### 11.2.2 Vícekroková registrace

V tomto případě je registrace do katalogu rozložena na několik kroků a v každém kroku se zadává část informací ohledně registrovaného webového projektu. Pokud se v žádném z kroků nenachází nějaká překážka, popsaná v následujících dvou kapitolách, je pravděpodobné, že data ze všech kroků vyhodnocuje skript na stejné URL adrese. Proto lze použít postup viz. Kapitola 11.2.1.

### 11.2.3 Strategie s captcha

Tato kapitola zastupuje všechny kombinace některé z předchozích strategií + záchytný obrázek, neboli captcha, jako bezpečnostní prvek v registračním formuláři katalogu.

Pokud by se implementovalo strojové čtení textu, mohli bychom taktéž využít postup viz. kapitola 11.2.1. Museli bychom ale brát v úvahu existenci mnoha typů záchytných obrázků (různé velikosti, barvy, fonty písma, jejich deformace či různé rušivé symboly, umístěné přes text, různá pozadí textu na záchytném obrázku atd.) a tím pádem nemožnost zaručit stoprocentní úspěšnost čtení (nebo o vícenásobných pokusech o přečtení). Jednak by bylo nutné z výstupu katalogu zjistit, že chyba byla právě v přečtení záchytného obrázku a pro takové případy použít jinou strategii, která by zahrnovala zásah uživatele.

Pro poloautomatizovaný proces lze využít strategii, popsanou v kapitole 11.2.6.

### **11.2.4 Strategie s výběrem kategorie**

Tato kapitola zastupuje všechny kombinace některé z předchozích strategií + výběr kategorie odkazu.

V případech, kdy katalog vyžaduje výběr kategorie nastává problém v podobě různých způsobů výběru kategorie u jednotlivých katalogů. Liší se nejen seznam kategorií, ale odlišnost je také v možnosti volby podkategorie (která je obvykle JavaScriptem upravena na základě výběru kategorie), HTML prvky pro výběr kategorie (select – resp. drop-down menu, přepínače, textové pole atd.). Postihnout efektivně co největší množství těchto odlišností by bylo předmětem dalšího průzkumu.

Pro poloautomatizovaný proces lze využít strategii, popsanou v kapitole 11.2.6.

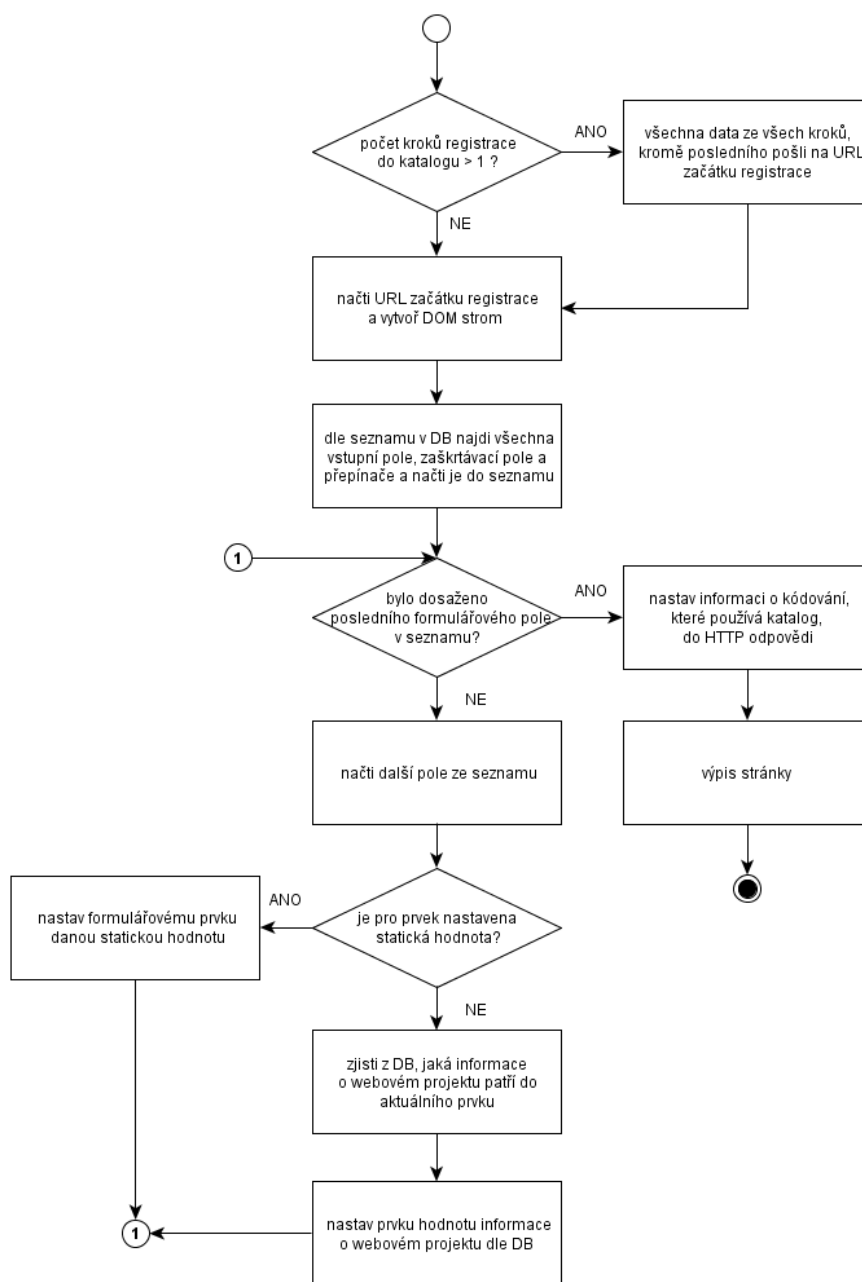
### **11.2.5 Jednoduchý registrační formulář, jež předchází výběr kategorie**

Toto je alternativou strategie z kapitoly 11.2.4., ale v tomto případě se výběr kategorie provádí vstupem do zvolené kategorie v katalogu a poté kliknutím na odkaz, který vede na začátek registrace. V tomto případě se identifikátor nebo název kategorie přenáší obvykle v URL adrese jako parametr.

Pro poloautomatizovaný proces lze využít strategii, popsanou v kapitole 11.2.6.

### **11.2.6 Předvyplňování polí formuláře pomocí PHP**

Tato strategie ulehčí uživateli vyplňování formulářů tím, že část formulářových polí bude předvyplněna. Mezi nevyplněná pole patří ověřovací kód (tzv. captcha), výběr kategorie. Ve výsledku bude prázdné také pole, kterému je JavaScriptem (součást načtené stránky z katalogu), po načtení stránky nastavena výchozí hodnota. Postup, jakým provádí skript předvyplnění a následnou prezentaci uživateli lze vyčíst z diagramu na obrázku 16.



**Obrázek 16: Postup při předvyplňování formulářových polí pomocí PHP**

### 11.2.7 Generování makra pro plug-in iMacros

Použití v případech, kdy nelze využít žádné z předchozích strategií. Taková situace nastává například když je použit HTTPS protokol, kontrola referrera (tedy předchozí navštívené stránky), registrační formulář je generován JavaScriptem, či v případech, kdy server generuje ID relace. Makro je spouštěno plug-inem webového prohlížeče – přehrání makra se tedy katalogu jeví jakoby celou registraci provedl člověk. Vygenerované makro stačí uložit do PC a následně jej v rozhraní plug-inu spustit, případně doplnit nevyplněné atributy.

Generování makra funguje na principu nahrazování názvů atributů (uzavřených ve složených závorkách) v makru, přidělenému ke katalogu, reálnými daty o webovém projektu.

### 11.3 Specializované typy strategií

Jako specializované strategie lze označit takové strategie, které lze aplikovat při registraci do katalogů, které využívají některé z katalogových aplikací. Tyto aplikace jsou speciální případy IS pro katalogy webových stránek. Myšlenka vychází z předpokladu, že katalog, založený na některém z vybraných typů IS pro katalogy, bude mít postup registrace totožný jako ostatní katalogy, postavené na stejném IS pro katalogy. Můžeme také předpokládat, že počet katalogů, postavených na takových IS se rozšiřuje rychleji, než u ostatních „jednoúčelových“ typů, protože jejich zprovoznění není tak náročné, jako vytvoření zcela nového systému. Z tohoto důvodu je vhodné se na ně zaměřit.

Zatím zná Registrátor 4 základní enginy, zaujímající ze všech bezmála 1900 aktivních katalogů v databázi přibližně 34%. Jedná se o tyto enginy:

- eSyndiCat (verze s nutností vytvořit před registrací uživatelský účet a druhá bez)
- Free – Globes (verze s nutností vytvořit před registrací uživatelský účet a druhá bez)
- PHP Link Directory
- Link Up

První dva enginy z předchozího seznamu existují ve verzi s nutností vytvoření uživatelského účtu (tedy pro použití strategie, popsané v kapitole 11.1.3) a druhá verze s registrací v podobě vyplnění formuláře s výběrem kategorie a opsáním kódu v záchytném obrázku (popsáno v kapitolách 11.2.3 a 11.2.4).

Dále by bylo možné seznam rozšířit o katalogy PR článků, kde naprostá většina funguje na enginu WordPress, který také vyžaduje vytvoření uživatelského účtu.



## 12 Závěr

Cílem této bakalářské práce bylo seznámit se s problematikou internetových katalogů, s principy jejich registrace a zmapování možností a technologií, kterými by bylo možné dosáhnout automatické nebo poloautomatické registrace do katalogů a vyhledávačů a implementovat aplikaci, která by využila těchto znalostí. Navrhl jsem tedy aplikaci s několika algoritmy, které umožní plné či částečné zautomatizování registrace do katalogu. Dále parser iMaker, plnící roli částečné automatizace získání dat o katalogu, potřebných k registraci a algoritmus detekce enginů, by při dopracování strategií registrací pro jednotlivé enginy, mohl výrazně snížit množství nutných zásahů ze strany obsluhy především v části přidávání nových katalogů do databáze.

Navrhnout systém, který by dokázal všechny katalogy, jejich různé kombinace použitých technologií a postupů registrace, zpracovat, by se musel vůči katalogu chovat jako člověk. To je způsobeno nejednotným postupem při registraci a různým druhům zabezpečení jako např. Záchytné obrázky. Z těchto důvodů je stále v nemalé míře nutný zásah lidské obsluhy.

### **Výhody Registrátoru oproti konkurenci:**

- několik strategií registrace, pokrývající větší procento katalogů poloautomatickým průběhem registrace a zahrnující také strategie se zcela automatickým procesem
- detekce katalogových enginů.

Adresářová struktura Registrátoru, použití jmenných prostorů, členění do modulů a také struktura databáze ponechává do budoucna možnost rozšíření o další nástroje, nebo zakomponování Registrátoru do jiného systému.

Jedním z bodů zadání bylo sledování vývoje „síly webu“ - hodnoty indexů page rank, srank, alexa apod. Rozsahem by tato část mohla být samostatnou bakalářskou prací.

### 13 Literatura, zdroje dat a zdrojových kódů

- [1] GRAPPONE, Jennifer a Gradiva COUZIN. *SEO: search engine optimization : ovládněte SEO a získejte výhodu před konkurencí : optimalizujte své webové stránky pro vyhledávací servery : přiveďte na své stránky zákazníky dříve, než to udělá konkurence*. Vyd. 1. Překlad Roman Skřivánek, Dana Balaščíková. Brno: Zoner Press, 2007, 328 s. ISBN 978-80-86815-85-5.
- [2] VONDRÁK CSc., Prof. Ing. Ivo. Úvod do softwarového inženýrství [online]. Ostrava, 2002. 74 s. Oborová práce. VŠB-TUO, Fakulta elektrotechniky a informatiky. Dostupné z: [http://vondrak.cs.vsb.cz/download/Uvod\\_do\\_softwaroveho\\_inzenyrstvi.pdf](http://vondrak.cs.vsb.cz/download/Uvod_do_softwaroveho_inzenyrstvi.pdf)
- [3] THE PHP GROUP. *Online PHP API dokumentace* [online]. 2001, 2012 [cit. 2012-04-24]. Dostupné z: <http://php.net>
- [4] Status Code Definitions. THE WORLD WIDE WEB CONSORTIUM. *W3C* [online]. 1999 [cit. 2012-04-24]. Dostupné z: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
- [5] PECKA, Miroslav. *Regulární výrazy* [online]. 2005, 2008 [cit. 2012-04-24]. Dostupné z: <http://www.regularnivyrazy.info>
- [6] IMacros TAG. IOPUS. *IMacro plug-in* [online]. 2000, 2012 [cit. 2012-04-24]. Dostupné z: <http://wiki.imacros.net/TAG>
- [7] SEZNAM KATALOGŮ.CZ. Seznam katalogů [online]. 2010, 2011 [cit. 2012-04-24]. Dostupné z: <http://www.seznamkatalogu.cz>
- [8] NETTE FOUNDATION, *Nette Framework* [online]. 2011 [cit. 2011-04-24]. Dostupné z: <http://nette.org/cs/>
- [9] <http://addons.nette.org/cs/gridito>, *Tabulkový doplněk Gridito pro Nette Framework* [online]. 2011 [cit. 2011-04-24]. Dostupné z: <http://addons.nette.org/cs/gridito>
- [10] NETTE FOUNDATION, *DiBi Framework* [online]. 2011 [cit. 2011-04-24]. Dostupné z: <http://www.dibi.com>
- [11] <http://simplehtmldom.sourceforge.net>, *Simple HTML DOM Parser* [online]. 2011 [cit. 2011-04-24]. Dostupné z: <http://simplehtmldom.sourceforge.net>
- [12] EASYPHP, *EasyPHP - Apache, PHP, MySQL, PHPMyAdmin* [online] 2011 [cit. 2011-04-24]. Dostupné z: <http://www.easyphp.org>
- [13] ORACLE CORPORATION, *NetBeans* [online]. 2011 [cit. 2011-04-24]. Dostupné z: <http://netbeans.org>
- [14] QUEST SOFTWARE, *Toad Data Modeler* [online]. 2011 [cit. 2011-04-24]. Dostupné z: <http://www.quest.com/toad-data-modeler/>
- [15] *Informační systém* In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 3. 11. 2007, last modified on 9. 3. 2011 [cit. 2011-04-12]. Dostupné z: [http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD\\_syst%C3%A9m](http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD_syst%C3%A9m)

## **14 Seznam příloh umístěných na CD**

- 1) Zdrojové kódy projektu Registrátor, včetně všech souborů použitých knihoven a frameworků v archivu „REGISTRATOR.zip“
- 2) Kompletní struktura databázových tabulek včetně všech dat ve formě SQL skriptu v archivu „DB.zip“
- 3) Jednorázové parsery v archivu „parsery.zip“